

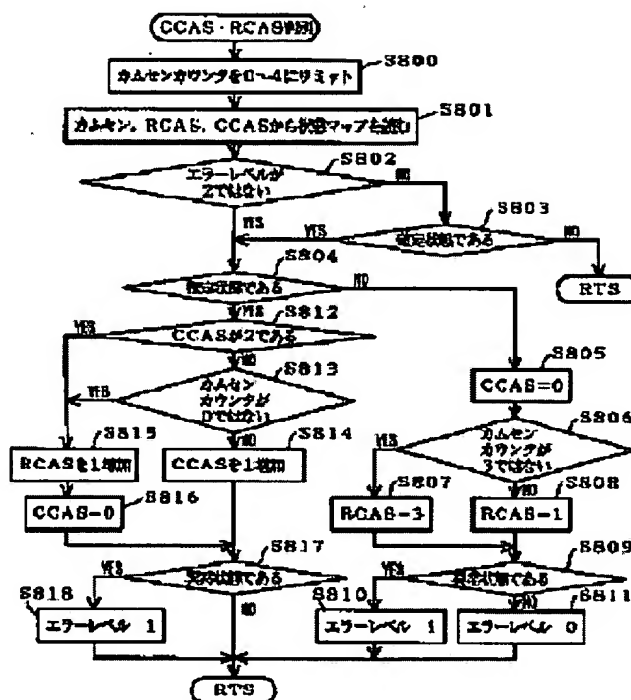
ENGINE CRANK POSITION DISCRIMINATING METHOD

Patent number: JP6213057
Publication date: 1994-08-02
Inventor: KURIHARA MASARU
Applicant: FUJI HEAVY IND LTD
Classification:
 - International: F02D45/00
 - european:
Application number: JP19930005335 19930114
Priority number(s):

Abstract of JP6213057

PURPOSE: To easily grasp an engine crank position varying according to control in simplifying a program by discriminating each state of a cylinder to come next and the crank position through crank position information, cylinder information or the like.

CONSTITUTION: A crank position in common with each cylinder, combustion sequence of cylinders and each discriminating state of the crank position inclusive of the combustion sequence of cylinders, each cylinder and the crank position are all found out by a crank position information variable (CCAS) calculated every prescribed crank position of an engine, a cylinder information variable (RCAS), integrated position information and an error level, thereby discriminating each state of a cylinder and a crank position to come next. First of all, a cam center counter is limited to 0 to 4 by a step S800. Next, proceeding to a step 801, whether all states causable in each combination from the cam center counter, RCAS and CCAS are normal or not as well as whether they are defined and to be presumed or not are all read, whereby the present state is evaluated, and such a state to be taken next is discriminated.



Data supplied from the esp@cenet database - Worldwide

Best Available Copy

THIS PAGE BLANK (USPTO)

(19)日本国特許庁(J P)

(12) 公開特許公報(A)

(11)特許出願公開番号

特開平6-213057

(43)公開日 平成6年(1994)8月2日

(51)Int.Cl.⁵

F 0 2 D 45/00

識別記号

3 6 2 A 7536-3G

庁内整理番号

F I

技術表示箇所

審査請求 未請求 請求項の数 4 O L (全 26 頁)

(21)出願番号 特願平5-5335

(22)出願日 平成5年(1993)1月14日

(71)出願人 000005348

富士重工業株式会社

東京都新宿区西新宿一丁目7番2号

(72)発明者 栗原 優

東京都三鷹市大沢3丁目9番6号 株式会

社スバル研究所内

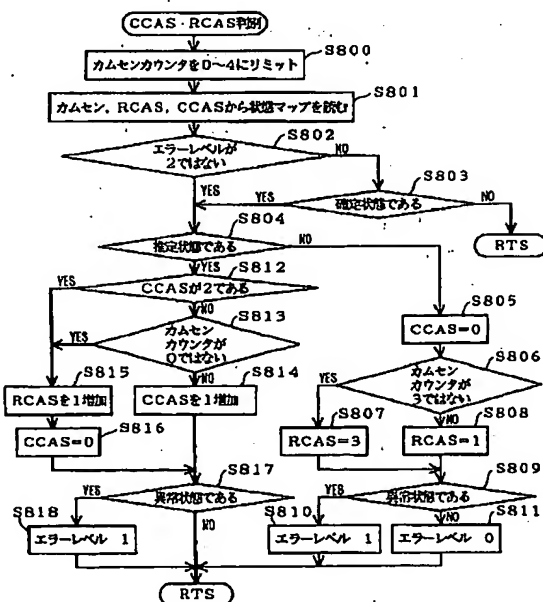
(74)代理人 弁理士 伊藤 進

(54)【発明の名称】 エンジンのクランク位置判別方法

(57)【要約】

【目的】 クランク位置判別のためのプログラムを簡略化してメモリ容量の増大や処理速度の低下を抑え、刻々と変化するエンジンのクランク位置の把握を容易にする。

【構成】 カムセンカウンタの計数値、気筒情報変数RCAS、クランク位置情報変数CCASにより、気筒・クランク位置状態マップCCHMAPから、クランク位置情報変数CCAS、カムセンカウンタ、及び気筒情報変数RCASの各組合わせの起こり得る全ての状態に対して正常か異常か確定して良いか推定すべきかを示す状態データを読み(S801)、今回のクランク位置変数CCAS、RCAS、ACAS、ECASを算出して刻々と変化するクランク位置の状態を把握するとともに、現在の状態を評価し次にとるべき状態を知る。これにより、プログラムを簡略化し、メモリ容量の節約、処理速度の向上を図るとともに、プログラムの可読性を向上して仕様変更に対しても柔軟に対処する。



【特許請求の範囲】

【請求項1】 エンジンのクランク位置検出信号によって起動するクランク同期割込み処理により、各気筒に共通したクランク位置を表わすクランク位置情報と、気筒の燃焼順序を表わす気筒情報と、気筒の燃焼順序及びクランク位置を表わす総合位置情報と、気筒及びクランク位置の判別状態を表わすエラーレベルとを算出し、上記クランク位置情報と上記気筒情報と上記総合位置情報と上記エラーレベルとにより、次にくるべき気筒及びクランク位置の状態を判別することを特徴とするエンジンのクランク位置判別方法。

【請求項2】 上記クランク位置情報を、各気筒の圧縮上死点前後で、初期値から所定値まで上記クランク位置検出信号の入力毎に変化する変数として算出するとともに、上記気筒情報を、気筒の燃焼順に従って気筒毎に変化する変数として算出し、さらに、上記総合位置情報を、上記クランク位置情報と上記気筒情報とからエンジンの1燃焼サイクル分に対応して周期的に変化する変数として算出し、上記エラーレベルを、気筒及びクランク位置の判別結果が確証をもった状態か不安の残る推定状態か不明の状態かに応じて変化する変数として算出することを特徴とする請求項1記載のエンジンのクランク位置判別方法。

【請求項3】 上記クランク位置検出信号の入力間におけるカム角センサ信号入力回数と、上記クランク位置情報と、上記気筒情報とによって決定される気筒及びクランク位置の状態データをストアした状態マップを設け、上記クランク同期割込み処理の起動時に、上記状態マップを参照して前回までの気筒及びクランク位置の状態を評価し、今回の上記クランク位置情報及び上記気筒情報を算出することを特徴とする請求項1記載のエンジンのクランク位置判別方法。

【請求項4】 上記状態データを、正常状態、異常状態、確定状態、及び、推定状態の各状態をコード化して組合せたデータとすることを特徴とする請求項3記載のエンジンのクランク位置判別方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、クランク位置の変化を容易に把握することのできるエンジンのクランク位置判別方法に関する。

【0002】

【従来の技術】 近年、自動車などの車輛にマイクロコンピュータが導入され、エンジン、パワートレインなどを高精度に制御することが可能となった。これにより、車輛制御システムの開発においては、マイクロコンピュータのソフトウェア開発が大きな比重を占めるようになり、制御アルゴリズム上の処理の効率化が重要な課題となっている。

【0003】 上記マイクロコンピュータによるエンジン制御システムにおいては、吸入空気量の算出、燃料噴射量の設定、点火時期の設定などといった所定時間毎の処理に加えて所定クランク角毎に実行すべき数多くの処理があり、この所定クランク角毎の処理は、例えば、特開昭53-40105号公報などに開示されているように、回転に同期した同期パルスにて優先順位が下位レベルの処理に対して割込みをかけるようになっている。

【0004】

【発明が解決しようとする課題】 ところで、刻々と変化するエンジンのクランク位置を判別して制御を適切に行なうためには、現在のクランク位置の状態から次の状態を把握し、また、判別したクランク位置が妥当なものであるのかを認識する必要がある。

【0005】 しかしながら、従来、このクランク位置を判別するためには、プログラム上で膨大な条件判断を行なわなければならない、メモリ容量の増大、処理速度の低下を招く一因ともなっていた。さらには、複雑な条件判断によってプログラムの可読性が悪化し、仕様変更の際にも、柔軟に対処することが困難であった。

【0006】 本発明は上記事情に鑑みてなされたもので、クランク位置判別のためのプログラムを簡略化してメモリ容量の増大や処理速度の低下を抑え、刻々と変化するエンジンのクランク位置を容易に把握することができるエンジンのクランク位置判別方法を提供することを目的としている。

【0007】

【課題を解決するための手段】 第1の発明は、エンジンのクランク位置検出信号によって起動するクランク同期割込み処理により、各気筒に共通したクランク位置を表わすクランク位置情報と、気筒の燃焼順序を表わす気筒情報と、気筒の燃焼順序及びクランク位置を表わす総合位置情報と、気筒及びクランク位置の判別状態を表わすエラーレベルとを算出し、上記クランク位置情報と上記気筒情報と上記総合位置情報と上記エラーレベルとにより、次にくるべき気筒及びクランク位置の状態を判別することを特徴とする。

【0008】 第2の発明は、第1の発明において、上記クランク位置情報を、各気筒の圧縮上死点前後で、初期値から所定値まで上記クランク位置検出信号の入力毎に変化する変数として算出するとともに、上記気筒情報を、気筒の燃焼順に従って気筒毎に変化する変数として算出し、さらに、上記総合位置情報を、上記クランク位置情報と上記気筒情報とからエンジンの1燃焼サイクル分に対応して周期的に変化する変数として算出し、上記エラーレベルを、気筒及びクランク位置の判別結果が確証をもった状態か不安の残る推定状態か不明の状態かに応じて変化する変数として算出することを特徴とする。

【0009】 第3の発明は、第1の発明において、上記クランク位置検出信号の入力間におけるカム角センサ信

号入力回数と、上記クランク位置情報と、上記気筒情報とによって決定される気筒及びクランク位置の状態データをストアした状態マップを設け、上記クランク同期割り込み処理の起動時に、上記状態マップを参照して前回までの気筒及びクランク位置の状態を評価し、今回の上記クランク位置情報及び上記気筒情報を算出することを特徴とする。

【0010】第4の発明は、第3の発明において、上記状態データを、正常状態、異常状態、確定状態、及び、推定状態の各状態をコード化して組合せたデータとする 10 ことを特徴とする。

【0011】

【作用】第1の発明では、エンジンの所定クランク位置毎に算出されるクランク位置情報と気筒情報と総合位置情報とエラーレベルとにより、各気筒に共通したクランク位置、気筒の燃焼順序、気筒の燃焼順序を含んだクランク位置、気筒及びクランク位置の判別状態を知り、次 15 にくるべき気筒及びクランク位置の状態を判別する。

【0012】第2の発明では、第1の発明におけるクランク位置情報と気筒情報とを、それぞれ、各気筒の圧縮 20 上死点前後で、初期値から所定値までクランク位置検出信号の入力毎に変化する変数、気筒の燃焼順に従って気筒毎に変化する変数として算出し、これらの変数から総合位置情報を算出する。また、エラーレベルを、各情報の判別結果が確証をもった状態か不安の残る推定状態か不明の状態かに応じて変化する変数として算出する。

【0013】第3の発明では、第1の発明において算出したクランク位置情報と気筒情報、及び、クランク位置検出信号の入力間におけるカム角センサ信号入力回数によ 30 って、気筒及びクランク位置の状態を決定し、その状態データを状態マップにストアする。そして、この状態データを参照することにより、前回までの気筒及びクランク位置の状態を評価し、今回の上記クランク位置情報及び上記気筒情報を算出する。

【0014】第4の発明では、第3の発明における状態データを、正常状態、異常状態、確定状態、及び、推定状態の各状態をコード化して組合せたデータとし、このデータにより、前回までの気筒及びクランク位置の状態を評価し、今回の上記クランク位置情報及び上記気筒情報を算出する。

【0015】

【実施例】以下、図面を参照して本発明の実施例を説明する。図面は本発明の一実施例に係り、図1はCCAS・RCAS判別サブルーチンのフローチャート、図2は 0.5ms毎の定期割り込み処理のフローチャート、図3はクラセン割り込み処理のフローチャート、図4はジョブ優先処理のフローチャート、図5はジョブ実行サブルーチンの部分フローチャート1、図6はジョブ実行サブ 40 ルーチンの部分フローチャート2、図7はジョブ実行サブルーチンの部分フローチャート3、図8はジョブ実行サ

ブルーチンの部分フローチャート4、図9はクランク位置算出サブルーチンのフローチャート、図10は半回転時間推定サブルーチンのフローチャート、図11はクラセンタイマのオーバーフロー割り込み処理のフローチャート、図12はジョブの実行状態を示す説明図、図13はジョブフラグの説明図、図14はクランク位置変数の説明図、図15はジョブ実行中フラグとオーバーラップカウンタの変化を示す説明図、図16はシステムシフトバッファの説明図、図17はクラセン間隔テーブルの説明図、図18は気筒・クランク位置状態マップの説明図、図19はクランク位置とエンジンの行程を示すタイムチャート、図20はエンジン系の概略構成図、図21はクランクロータとクランク角センサの正面図、図22はカムロータとカム角センサの正面図、図23は電子制御系の回路構成図である。

【0016】本実施例のエンジン制御システムでは、図23に示すマイクロコンピュータを中核とした電子制御装置(ECU)50により図20に示すエンジン系が制御され、燃料噴射制御、点火時期制御などが行なわれ 50 る。上記ECU50のマイクロコンピュータには、新しい概念に基づくオペレーティングシステム(OS)が搭載され、このOSにより、各センサ類からの信号入力処理、エンジン回転数算出処理、吸入空気量算出処理、燃料噴射量設定処理、点火時期設定処理などといった各制御項目毎のジョブが管理されて効率的に実行されるようになっている。

【0017】まず、上記ECU50によって制御されるエンジン系の機器構成について説明する。

【0018】図20に示すように、エンジン1(図においては水平対向4気筒型エンジンを示す)は、シリンダヘッド2の吸気ポート2aにインテークマニホルド3が連通され、このインテークマニホルド3の上流にエアチャンバ4を介してスロットル通路5が連通されている。このスロットル通路5の上流側には、吸気管6を介してエアクリーナ7が取付けられ、このエアクリーナ7が吸入空気の取り入れ口であるエアインテークチャンバ8に連通されている。

【0019】また、上記排気ポート2bにエキゾーストマニホルド9を介して排気管10が連通され、この排気管10に触媒コンバータ11が介装されてマフラ12に連通されている。一方、上記スロットル通路5にスロットルバルブ5aが設けられ、このスロットル通路5の直上流の上記吸気管6にインタークーラ13が介装され、さらに、上記吸気管6の上記エアクリーナ7の下流側にレゾネータチャンバ14が介装されている。

【0020】また、上記レゾネータチャンバ14と上記インテークマニホルド3とを連通して上記スロットルバルブ5aの上流側と下流側とをバイパスするバイパス通路15に、アイドルスピードコントロールバルブ(ISCV)16が介装されている。さらに、このISCV1

6の直下流側に、吸気圧が負圧のとき開弁し、またターボチャージャ18によって過給されて吸気圧が正圧になったとき閉弁するチェックバルブ17が介装されている。

【0021】上記ターボチャージャ18は、上記吸気管6の上記レゾネータチャンバ14の下流側にコンプレッサハウジングが介装され、タービンハウジングが上記排気管10に介装されている。さらに、上記ターボチャージャ18のタービンハウジング流入口には、ウエストゲート弁19が介装され、このウエストゲート弁19に

は、ウエストゲート弁作動用アクチュエータ20が連設されている。

【0022】上記ウエストゲート弁作動用アクチュエータ20は、ダイヤフラムにより2室に仕切られ、一方がウエストゲート弁制御用デューティソレノイド弁21に連通される圧力室を形成し、他方が上記ウエストゲート弁19を閉方向に付勢するスプリングを収納したスプリング室を形成している。

【0023】上記ウエストゲート弁制御用デューティソレノイド弁21は、上記レゾネータチャンバ14と上記吸気管6の上記ターボチャージャ18のコンプレッサハウジング下流とを連通する通路に介装されており、ECU50から出力される制御信号のデューティ比に応じて、上記レゾネータチャンバ14側の圧力と上記コンプレッサハウジング下流側の圧力とを調圧し、上記ウエストゲート弁作動用アクチュエータ20の圧力室に供給する。

【0024】すなわち、上記ECU50によって上記ウエストゲート弁制御用デューティソレノイド弁21を制御し、上記ウエストゲート弁作動用アクチュエータ20を作動させて上記ウエストゲート弁19による排気ガスリリーフを調整することにより、上記ターボチャージャ18による過給圧を制御するようになっている。

【0025】また、上記インテークマニホルド3に絶対圧センサ22が通路23を介して連通され、この通路23に、上記絶対圧センサ22と上記インテークマニホルド3あるいは大気とを選択的に連通する吸気管圧力／大気圧切換ソレノイド弁24が介装されている。

【0026】さらに、上記インテークマニホルド3の各気筒の各吸気ポート2aの直上流側にインジェクタ25が臨まされ、また、上記シリンダヘッド2の各気筒毎に、その先端を燃焼室に露呈する点火プラグ26aが取付けられ、この点火プラグ26aに連設する点火コイル26bにイグナイタ27が接続されている。

【0027】上記インジェクタ25には、燃料タンク28内に設けたインタンク式の燃料ポンプ29から燃料フィルタ30を経て燃料が圧送され、プレッシャレギュレータ31にて調圧される。

【0028】また、上記吸気管6の上記エアークリーナ7の直下流に、ホットワイヤ式あるいはホットフィルム

式などの吸入空気量センサ32が介装され、上記スロットルバルブ5aに、スロットル開度センサ33aとアイドルスイッチ33bとを内蔵したスロットルセンサ33が連設されている。

【0029】さらに、上記エンジン1のシリンダブロック1aにノックセンサ34が取付けられるとともに、このシリンダブロック1aの左右両バンクを連通する冷却水通路35に冷却水温センサ36が臨まされ、上記排気管10の上記エキゾーストマニホルド9の集合部にO2センサ37が臨まされている。

【0030】また、上記シリンダブロック1aに支承されたクランクシャフト1bにクランクロータ38が軸着され、このクランクロータ38の外周に、電磁ピックアップなどからなるクランク角センサ39が対設されている。さらに、上記エンジン1のカムシャフト1cに連設するカムロータ40に、電磁ピックアップなどからなる気筒判別用のカム角センサ41が対設されている。尚、上記クランク角センサ39及び上記カム角センサ41は、電磁ピックアップなどの磁気センサに限らず、光センサなどでも良い。

【0031】上記クランクロータ38は、図21に示すように、その外周に突起38a、38b、38cが形成され、これらの各突起38a、38b、38cが、各気筒(#1、#2と#3、#4)の圧縮上死点前(BTDC) $\theta 1, \theta 2, \theta 3$ の位置に形成されており、本実施例においては、 $\theta 1 = 97^\circ \text{ CA}$ 、 $\theta 2 = 65^\circ \text{ CA}$ 、 $\theta 3 = 10^\circ \text{ CA}$ である。

【0032】上記クランクロータ38の各突起は、上記クランク角センサ39によって検出され、図19に示すように、BTDC 97° 、 65° 、 10° のクランクバルスがエンジン1/2回転毎(180° CA 毎)に出力される。そして、上記クランク角センサ39からのクランク位置検出信号の入力間隔時間がタイマによって計時され、エンジン回転数が算出される。

【0033】尚、突起38bは、点火時期設定の際の基準クランク角となり、また、突起38cは、始動時噴射開始時期の基準クランク角となるとともに始動時の固定点火時期を示すクランク角となる。

【0034】また、図22に示すように、上記カムロータ40の外周には、気筒判別用の突起40a、40b、40cが形成され、突起40aが#3、#4気筒の圧縮上死点後(ATDC) $\theta 4$ の位置に形成され、突起40bが3個の突起で構成されて最初の突起が#1気筒のATDC $\theta 5$ の位置に形成されている。さらに、突起40cが2個の突起で形成され、最初の突起が#2気筒のATDC $\theta 6$ の位置に形成されている。本実施例においては、 $\theta 4 = 20^\circ \text{ CA}$ 、 $\theta 5 = 5^\circ \text{ CA}$ 、 $\theta 6 = 20^\circ \text{ CA}$ である。

【0035】そして、上記カムロータ40の各突起が上記カム角センサ41によって検出され、各気筒の燃焼行

手順を#1→#3→#2→#4とした場合、この燃焼行程順と、上記カム角センサ41からのカムパルスをカウンタによって計数した値とのパターン(図19参照)に基づいて、気筒判別がなされる。

【0036】一方、図23に示すECU50は、燃料噴射制御、点火時期制御などを行なうメインコンピュータ51と、ノック検出処理を行なう専用のサブコンピュータ52との2つのコンピュータを中心として構成され、各部に所定の安定化電源を供給する定電圧回路53や各種の周辺回路が組込まれている。

【0037】上記定電圧回路53は、ECUリレー54のリレー接点を介してバッテリー55に接続され、このバッテリー55に、上記ECUリレー54のリレーコイルがイグニッションスイッチ56を介して接続されている。また、上記バッテリー55には、上記定電圧回路53が直接接続され、さらに、燃料ポンブリレー57のリレー接点を介して燃料ポンプ29が接続されている。

【0038】すなわち、上記定電圧回路53は、上記イグニッションスイッチ56がONされ、上記ECUリレー54のリレー接点が閉となったとき、制御用電源を供給し、また、上記イグニッションスイッチ56がOFFされたとき、バックアップ用の電源を供給する。

【0039】上記メインコンピュータ51は、CPU58(以下、メインCPU58と称する)、ROM59、RAM60、上記イグニッションスイッチ56がOFFされたときにも上記定電圧回路53からバックアップ電源が供給されてデータを保持するバックアップRAM61、カウンタ・タイマ群62、シリアル通信インターフェースであるSCI63、及び、I/Oインターフェース64がバスライン65を介して接続されたマイクロコンピュータである。

【0040】尚、上記カウンタ・タイマ群62は、フリーランカウンタ、カム角センサ(以下、適宜、カムセンと略記する)信号の入力計数用カムセンカウンタなどの各種カウンタ、燃料噴射タイマ、点火タイマ、後述する0.5ms毎の定期割込みを発生させるための定期割込みタイマ、クランク角センサ(以下、適宜、クラセンと略記する)信号の入力間隔計時用クラセンタイマ、及び、システム異常監視用のウォッチドッグタイマなどの各種タイマを便宜上総称するものであり、上記メインコンピュータ51においては、その他、各種のソフトウェアカウンタ・タイマが用いられる。

【0041】また、上記サブコンピュータ52も、上記メインコンピュータ51と同様、CPU71(以下、サブCPU71と称する)、ROM72、RAM73、カウンタ・タイマ群74、SCI75、及び、I/Oインターフェース76がバスライン77を介して接続されたマイクロコンピュータであり、上記メインコンピュータ51とサブコンピュータ52とは、上記SCI63、75を介してシリアル通信ラインにより互いに接続されて

いる。

【0042】上記メインコンピュータ51のI/Oインターフェース64には、入力ポートに、吸入空気量センサ32、スロットル開度センサ33a、水温センサ36、O₂センサ37、絶対圧センサ22、車速センサ42、及び、バッテリー55が、8チャンネル入力A/D変換器66を介して接続されるとともに、アイドルスイッチ33b、クランク角センサ39、カム角センサ41が接続されており、さらに、始動状態を検出するためにスタータスイッチ43が接続されている。

【0043】尚、本実施例においては、上記A/D変換器66は、7チャンネル分の入力を使用され、残りの1チャンネルは予備となっている。

【0044】また、上記I/Oインターフェース64の出力ポートには、イグニタ27が接続され、さらに、駆動回路67を介して、ISCV16、インジェクタ25、燃料ポンブリレー57のリレーコイル、および、ウエストゲート弁制御用デューティソレノイド弁21、吸気管圧力/大気圧切換ソレノイド弁24が接続されている。

【0045】一方、上記サブコンピュータ52のI/Oインターフェース76は、入力ポートに、クランク角センサ39、カム角センサ41が接続されるとともに、A/D変換器78、周波数フィルタ79、アンプ80を介してノックセンサ34が接続されており、上記ノックセンサ34からのノック検出信号が上記アンプ80で所定のレベルに増幅された後に上記周波数フィルタ79により必要な周波数成分が抽出され、上記A/D変換器78にてデジタル信号に変換されて入力されるようになっている。

【0046】上記メインコンピュータ51では、各センサ類からの検出信号を処理し、燃料噴射パルス幅、点火時期などを演算する。すなわち、吸入空気量センサ32の出力信号から吸入空気量を算出し、RAM60及びバックアップRAM61に記憶されている各種データに基づき、吸入空気量に見合った燃料噴射量を演算し、また、点火時期などを算出する。

【0047】そして、上記燃料噴射量に相応する駆動パルス幅信号を、駆動回路67を介して所定のタイミングで該当気筒のインジェクタ25に出力して燃料を噴射し、また、所定のタイミングでイグニタ27に点火信号を出力し、該当気筒の点火プラグ26aを点火する。

【0048】その結果、該当気筒に供給された混合気が爆発燃焼し、エキゾーストマニホールド9の集合部に臨まされたO₂センサ37により排気ガス中の酸素濃度が検出され、この検出信号が波形整形された後、上記メインCPU58で基準電圧(スライスレベル)と比較され、エンジンの空燃比状態が目標空燃比に対し、リッチ側にあるか、リーン側にあるかが判別され、空燃比が目標空燃比となるようフィードバック制御される。

【0049】一方、上記サブコンピュータ52では、エンジン回転数とエンジン負荷とに基づいてノックセンサ34からの信号のサンプル区間を設定し、このサンプル区間でノックセンサ34からの信号を高速にA/D変換して振動波形を忠実にデジタルデータに変換し、ノック発生の有無を判定する。

【0050】上記サブコンピュータ52のI/Oインターフェース76の出力ポートは、上記メインコンピュータ51のI/Oインターフェース64の入力ポートに接続されており、上記サブコンピュータ52でのノック判定結果がI/Oインターフェース76に出力される。そして、上記メインコンピュータ51では、上記サブコンピュータ52からノック発生有りの判定結果が出力されると、SCI63を介してシリアル通信ラインよりノックデータを読み込み、このノックデータに基づいて直ちに該当気筒の点火時期を遅らせ、ノックを回避する。

【0051】このようなエンジン制御において、上記メインコンピュータ51では、各センサ類からの信号入力処理、エンジン回転数算出、吸入空気量算出、燃料噴射量算出、点火時期算出といった各項目毎の各種プログラムが、一つのOSの管理下で効率的に実行される。このOSは、車輛制御のための各種マネジメント機能、及び、このマネジメント機能に密着した内部ストラテジーを有し、各種ジョブを体系的に結合する。

【0052】上記OSのマネジメント機能としては、

(1-1)ジョブの優先処理

(1-2)セクション定義による各ジョブの分割ファイル対応

(1-3)スタックの使用状況モニタ機能

(1-4)異常割込み動作のモニタ機能

(1-5)ジョブ毎に固有の制約を作らない標準マップ・標準ワークメモリ設定

などの機能があり、制御ストラテジーの開発環境を向上させるとともに、限られたCPU能力を最大限に発揮させ、デジタル制御理論の基本である等時間間隔処理を可能な限り達成することができる。

【0053】等時間間隔処理としては、0.5ms毎の定期割込みを基本として、2, 4, 10, 50, 250ms毎の5種類の等間隔割込みジョブが用意されており、また、エンジン回転に同期した処理として、クランク角信号入力により即割込み実行される高優先クラセンジョブ（以下、単にクラセンジョブと称する）と、より優先順位が高い他のジョブがないときにクランク角信号入力により割込み実行される比較的緊急度の低い低優先クラセンジョブとが用意されている。

【0054】これらの各ジョブには、クラセンジョブ>2msジョブ>4msジョブ>10msジョブ>低優先クラセンジョブ>50msジョブ>250msジョブの順で、7~1の優先レベルが高位側から低位側に向かって付けられており、図12に示すように、高速ジョブに

対し低速ジョブが分割して処理されるとともに、各ジョブの多重待ち処理が行なわれる。

【0055】また、上記OSの下で働く各プログラムは、機能別の管理領域すなわちセクション領域毎に順番に配列されており、各セクション領域には機能毎にセクション宣言によって名前が付けられている。各ストラテジーファイル側で使用する主なセクション領域は、

○変数宣言領域

○自己ファイル名、フィル制作時の自動記録領域

○セッティングデータ領域

○クラセンジョブ領域

○2msジョブ領域

○4msジョブ領域

○10msジョブ領域

○低優先クラセンジョブ領域

○50msジョブ領域

○250msジョブ領域

○リセット時初期化ジョブ領域

○エンスト時初期化ジョブ領域

○バックグラウンドジョブ領域

○プログラム本体の領域

であり、機能毎にファイルを分割してプログラム開発が可能になるとともに、プログラムの構造化記述を可能にする。

【0056】また、上記OSには、以上のマネジメント機能に密着した内部ストラテジーとして、

(2-1)A/D変換処理

(2-2)クランク位置に係る各種情報の算出

(2-3)デバッグ用シュミレーション機能（エンジン回転及びA/D変換）

(2-4)点火タイマのセット

(2-5)燃料噴射タイマのセット

などの機能を備えており、さらに、これらの機能に係る各種サービスルーチンが各ジョブ中に用意されている。

【0057】従来、このような機能は各ジョブレベルで達成するようになっていたが、本システムにおいては、すべてOS側に用意され、OS側で処理したA/D変換結果、クランク位置情報、エンジン回転数などに基づいて、ユーザー側の各ジョブで、燃料噴射量、点火時期などを設定すると、これらの指示値がOSによって燃料噴射タイマ、点火タイマにセットされるようになっていく。

【0058】次に、上記メインコンピュータ51におけるジョブ処理の機能を、図1~図11のフローチャートに基づいて説明する。尚、サブコンピュータ52はノック検出処理専用のコンピュータであるため、その動作説明を省略する。

【0059】まず、イグニッションスイッチ56がONされてシステムに電源が投入されると、リセットに伴うリセット割込みが起動し、各種イニシャライズが行なわ

れるとともに、0.5ms毎に定期割込みを起動するための定期割込みタイマが起動され、クランク角センサ39からの信号入力毎(BTDC97°, 65°, 10°CA毎のエンジン1回転に6回)に起動されるクラセン割込みが許可され、その後、バックグランドジョブの実行状態となる。

【0060】そして、このバックグランドジョブの上で、0.5ms毎の定期割込みと、エンジン1回転に6回のクラセン割込みとにより、7レベルのジョブが優先処理される。この2つの割込みにおいては、各自の処理

を実行後、共通のアドレスにジャンプし、ジョブ優先処理を実行する。

【0061】尚、上記リセット割込みは、内部演算において0による除算を実行した場合や、無限ループが発生した場合など、正常時には発生しない要因によっても、起動される。

【0062】まず、図2に示す0.5ms毎の定期割込みについて説明する。この定期割込みでは、ステップS100で、OS用ワークエリアを設定し、ステップS101で、ウオッチドッグタイマを初期化すると、ステップS102へ

進んで、P-RUNフラグを20回に1回すなわち10ms毎に反転する。このP-RUNフラグは、図示しない保護回路によってシステムが自動的にリセットされないようにするためのフラグであり、システムが正常に動作して一定時間毎(10ms毎)に反転される限り、上記保護回路の作動が阻止される。

【0063】次いで、ステップS103へ進み、スイッチ出力の転写を行なう。このスイッチ出力は、各ジョブ中でメモリに書き込んだビットのON、OFF値であり、各

ジョブからは直接I/Oインターフェース64の出力ポートに出力せず、OS側で0.5ms毎にメモリの値を出力ポートに転写する。

【0064】次に、ステップS104へ進むと、A/D変換サブルーチンを実行してA/D変換に係る各種設定を行ない、ステップS105で、ジョブフラグ作成サブルーチンを実行して、2, 4, 10, 50, 250ms毎の各ジョブ割込み要求を示すジョブフラグJB_FLGを作成した後、ステップS106で、A/D変換をスタートする。

【0065】上記A/D変換は、基本的に、A/D変換器66の8チャンネル入力が0.5ms毎に所定の変換順番毎に処理され、4ms周期で全入力の変換が行なわれる。但し、特定の1つのチャンネルは、回転脈動が発生する吸入管圧力などをA/D変換するためクランク角90°毎に(0.5msの時間精度で)同期し、変換順番に対して割込んだ形で処理が行なわれ、その後の入力の順番を1つ遅れにする。

【0066】尚、エンジン回転数3750rpm以上では、A/D変換の最後の順番の入力が完全に停止し、7500rpm以上では、最後から2番目の入力も停止するが、A/D変換の順番は、スロットル開度、吸入空気

量など変化の速いものを先として、冷却水温、電圧など比較的变化の遅いものが後になるように設定してあり、且つ、最後のA/D変換順番をクランク同期入力に設定してあるため、特に支障は生じない。

【0067】また、図13に示すように、上記ジョブフラグJB_FLGは、1バイト変数の各ビットを各ジョブに対応するフラグとして割当てたものであり、複数のジョブ要求が同時に可能なようになっている。この1バイト変数のビット1~ビット7は優先レベル1~7に対応し、それぞれ、250msジョブ、50msジョブ、低優先クラセンジョブ、10msジョブ、4msジョブ、2msジョブ、クラセンジョブのフラグに割当てられている。そして、所定のビットが立てられたとき、対応する優先レベルのジョブ割込み要求がなされる。尚、ビット0はバックグランドジョブのフラグに割当てられて通常は参照されない。

【0068】そして、上記ステップS105でジョブフラグ作成サブルーチンによりジョブフラグJB_FLGを作成し、上記ステップS106でA/D変換をスタートした後は、ステップS107へ進み、ジョブフラグJB_FLGのいずれかのジョブに対応するビットが立っているかを調べる。

【0069】その結果、ジョブフラグJB_FLGのビットが一つも立っていないときには、どのジョブからも要求がないため割込みを終了し、ジョブフラグJB_FLGのいずれかのビットが立っているときには、ステップS108へ進んで、現状レベル(この定期割込みが実行される時点で所定の優先レベルのジョブが実行されていた状態)以下のフラグがないかを調べる。

【0070】上記ステップS108で、現状レベル以下のフラグがないときには、ラベルWAR_JBで示される図4のジョブ優先処理にジャンプし、現状レベル以下のフラグがあるときには、ステップS109で、現状レベル以下のレベルのオーバーラップカウンタOLCを1増加させる。

【0071】上記オーバーラップカウンタOLCは、ジョブ要求を記憶するためのカウンタであり、各優先レベル毎に1バイト割当てられ、上記ジョブフラグJB_FLGによるジョブ要求時にインクリメント、ジョブ終了時にデクリメントされる。すなわち、カウンタによってジョブ要求を記憶することによりジョブの多重要求に対応することができるのである。

【0072】次いで、上記ステップS109からステップS110へ進み、現状レベルより高いフラグがないかを調べ、現状レベルより高いフラグがないときには、ルーチンを抜けて割込みを終了し、現状レベルより高いフラグがあるときには、ラベルWAR_JBのジョブ優先処理へジャンプする。

【0073】一方、この0.5ms毎の定期割込みに対し、図3のクラセンによる割込みでは、ステップS200

10

20

30

40

50

で、OS用ワークエリアを設定すると、ステップS201で、後述するクランク位置・半回転時間算出のサブルーチンを実行し、現在のクランク位置を判別するためのクランク位置変数、及び、最新の3つのクラセン間隔の和であるエンジン半回転の経過時間すなわち半回転時間を算出する。

【0074】上記クランク位置変数は、OS中で用意されるシステム変数であり、図14に示すように、#1～#4気筒に対するクランク位置を、 97° 、 65° 、 10° CAによって12の状態に区分し、現在のクランク位置を表わす。

【0075】すなわち、各気筒毎に、0、1、2の数値でクラセン入力順を示すクランク位置情報変数S_{CCAS}、#1気筒を0、#3気筒を1、#2気筒を2、#4気筒を3として気筒の燃焼順を示す気筒情報変数S_{RCAS}、及び、0～11の数値でクラセン順序及び気筒順序を総合的に表わす総合位置変数S_{ACAS}の3変数によって現在のクランク位置を表わし、さらに、クランク位置が確証をもって正常に判別されたときを0、判別結果がつつまが合わず不安の残る推定状態を1、不明な状態を2とするエラーレベルS_{ECAS}により、クランク位置の判別状況を表わすようになっている。尚、図14においては、システム変数であることを示すS_—を省略している。

【0076】次いで、上記ステップS201からステップS202へ進むと、クランク位置・半回転時間算出のサブルーチンにおいてクランク位置判定が正常に終了したかあるいは判定不能であったかを、アキュムレータAにストアされているコードを読み出すことにより調べる（エラーコード1、正常終了コード0）。

【0077】そして、上記ステップS202で、アキュムレータAの値が1であり、クランク位置が判定不能であったときには、割込みを終了し、アキュムレータAの値が0であり、クランク位置が正常に判定されているときには、ステップS203へ進み、エンストフラグを解除する。

【0078】尚、上記エンストフラグは、エンジンがエンスト状態であることを示すフラグであり、クラセン間隔が0.5sec以上の時間（約30rpm以下）のとき、50msジョブに用意されているエンスト処理ルーチンによりセットされ、このクラセン割込みによりクリアされてエンスト状態が解除される。

【0079】次に、ステップS204へ進むと、点火タイムセットのサブルーチンを実行し、ユーザージョブ側で設定した点火時期の指示値に基づいて作成された点火スケジュールに従って点火タイムをセットする。この点火スケジュールは、ドエル開始時期、ドエルオン待ち時間、ドエルオフ待ち時間などをメンバーとする構造体変数であり、10msジョブ中に作成ルーチンが用意され、この点火スケジュールに従って点火シーケンスが決定される。

【0080】次いで、ステップS205で、燃料噴射タイマセットのサブルーチンを実行し、ユーザージョブ側で設定した燃料噴射量の指示値（各気筒毎の噴射幅）に対し、燃料噴射開始時期などを燃料噴射タイマにセットしてステップS206へ進む。

【0081】ステップS206では、このクラセンが実行された現状のジョブレベルが自身のジョブレベルであるかを判別し、現状がクラセンジョブ自身のレベルであるときには、ステップS207、S208で、クラセンジョブ、低優先クラセンジョブのオーバーラップカウンタOLCを、それぞれ1増加させて割込みを終了し、現状のジョブレベルがクラセンジョブのレベルでないときには、ステップS209で、現状のジョブレベルが低優先クラセンジョブのレベル以上であるかを調べる。

【0082】そして、現状のジョブレベルが低優先クラセンジョブ以上であるときには、上記ステップS209からステップS210へ進んで、低優先クラセンジョブのオーバーラップカウンタOLCを1増加させると、ステップS211で、クラセンジョブのジョブフラグをセットし、ラベルWAR_JBのジョブ優先処理へジャンプする。

【0083】一方、上記ステップS209で、現状のジョブレベルが低優先クラセンジョブ以上でないときには、上記ステップS209からステップS212へ進み、クラセンジョブのジョブフラグをセットすると、ステップS213で、低優先クラセンジョブのジョブフラグをセットし、ラベルWAR_JBのジョブ優先処理へジャンプする。

【0084】このジョブ優先処理では、ステップS300で、ジョブの優先レベルを示す1バイト変数であるジョブレベルJ_BLEVを1つ上げると、ステップS301へ進んで、この優先レベルに対応するジョブフラグが立っていないか調べる。そして、ジョブフラグが立っていないときには、ステップS300へ戻ってさらにジョブレベルJ_BLEVを1つ上げ、ジョブフラグが立っているときには、ステップS302へ進み、ジョブフラグの立っているジョブのオーバーラップカウンタOLCを初期値の0から1にし、ステップS303へ進む。

【0085】ステップS303では、より上のジョブフラグがあるかを調べ、より上のジョブがあるときには、ステップS300へ戻って前述の処理を繰り返し、より上のジョブがないときには、ステップS304へ進んで、ジョブ実行中フラグJ_BRUNをセットすると、ステップS305で、後述するジョブ実行サブルーチンにより最上位のジョブを実行する。

【0086】上記ジョブ実行中フラグJ_BRUNは、ジョブの実行開始時にセットされ、終了時にクリアされるフラグであり、このフラグにより、処理の途中で、より優先度の高いジョブによって割込まれたジョブを識別することができる。

【0087】例えば、図15に示すように、J_BLEV=4の10msジョブを実行中、J_BLEV=6の

2msジョブの割込み要求がなされると、10msジョブの処理が中断され、より優先度の高い2msジョブが、JB_RUN=1、OLC=1にセットされ、実行される。そして、この2msジョブの処理中に、JB_LEV=5の4msジョブの割込み要求が発生すると、この4msジョブは、JB_RUN=0、OLC=1とされて割込みが受けられるが、実行はされず待機状態となる。

【0088】その後、ジョブ実行サブルーチンによるジョブの実行が終了すると、上記ステップS305からステップS306へ進んでオーバーラップカウンタOLCを1減らし、ステップS307で、オーバーラップカウンタOLCがゼロになったか否かを調べる。その結果、オーバーラップカウンタOLCがゼロになっておらず、同じ優先レベルでジョブ割込み要求が複数回あるときには、ステップS305へ戻ってジョブを繰返し実行し、オーバーラップカウンタOLCがゼロになったとき、ステップS307からステップS308へ進んで、ジョブ実行中フラグJB_RUNをクリアする。

【0089】次に、ステップS309へ進み、ジョブレベルJB_LEVを1つ下げて次のジョブレベルに移ると、ステップS310で、このジョブレベルJB_LEVがゼロになったか否かを調べる。そして、ジョブレベルJB_LEVがゼロのときには、この割込みを終了し、ジョブレベルJB_LEVがゼロでないときには、ステップS311へ進んで、オーバーラップカウンタOLCがゼロか否かを調べる。

【0090】上記ステップS311で、オーバーラップカウンタOLCがゼロのときには、このレベルではジョブ要求はないため、上記ステップS311からステップS309へ戻って、ジョブレベルJB_LEVをさらに1つ下げて同様の処理を繰返し、オーバーラップカウンタOLCがゼロでないときには、ステップS312へ進んで、このジョブレベルにおいて、ジョブ実行中フラグJB_RUNがセットされているか否かを調べる。

【0091】上記ステップS312で、ジョブ実行中フラグJB_RUNがセットされているときには、割込み前にジョブを実行中であつたため、割込みを終了して割込み前のジョブへ戻り、ジョブ実行中フラグJB_RUNがセットされていないければ、ステップS304へ戻って、このレベルのジョブを実行し、同様の処理を繰返す。

【0092】すなわち、図15において、JB_LEV=6の2msジョブが終了し、OLC=0、JB_RUN=0になると、ジョブレベルが1つ下げられ、JB_LEV=5の4msジョブが、JB_RUN=0、OLC=1の待機状態からJB_RUN=1にセットされ、実行される。さらに、4msジョブが終了すると、JB_LEV=4に移り、JB_RUN=1（ジョブ実行中）の状態から、2msジョブ及び4msジョブによって中断されていた10msジョブの処理が再開される。

【0093】このように、0.5ms毎の定期割込み、クラセン割込みを基本タイミングとして、各ジョブの優先レベル及び実行タイミングを知らせるジョブフラグJB_FLGを作成するため、可能な限り正確に、等時間間隔処理、エンジン回転同期処理を実現し、各ジョブを効率良く処理することができる。さらに、基本タイミングとなる各割込み毎に更新されるジョブフラグJB_FLGによらず、オーバーラップカウンタOLCによってジョブの多重要求を記憶するため、あるジョブの処理時間が長引き、再度、同じジョブを実行すべきタイミングとなった場合においても、処理を途中で放棄することなく、可能な限り最後まで処理を継続することができる。

【0094】次に、図5～図8のジョブ実行サブルーチンについて説明する。

【0095】まず、ステップS500で、ジョブフラグJB_FLGを参照して実行すべきジョブがクラセンジョブでないか否かを調べ、クラセンジョブでないときには、ラベルALJ10へ分岐し、クラセンジョブのときには、ステップS501へ進んで、気筒判別がついているか否かを調べる。

【0096】そして、気筒判別がついていないときには、そのままルーチンを抜けてジョブを実行せず、気筒判別がついているとき、上記ステップS501からステップS502へ進んで、オーバーラップカウンタOLCの値を参照して多重待ち状態であるか否かを調べる。

【0097】上記ステップS502では、多重待ち状態でないとき、ステップS503へ進んで、クラセン割込み毎に算出されるシステム変数S_ACAS（クランク総合位置変数）をユーザー変数ACASとし、一方、多重待ち状態のときには、ステップS504へ分岐し、ユーザー変数ACASを一つ増やして12で割った剰余をとって新たなユーザー変数ACASとし、このユーザー変数ACASを0, 1, 2, ..., 11, 0, 1, ...とソフトウェア的に更新してゆく。

【0098】すなわち、クラセンジョブ及び低優先クラセンジョブは、自身または優先度の高いジョブに邪魔されて遅れることがあるが、クラセン割込みは正確にクランク角センサ信号に同期して実行され、システム変数S_ACASはジョブの遅れに関係なく更新される。

【0099】従って、ジョブ中でシステム変数S_ACASを参照して気筒及びクランク位置に係る情報を知り、この情報に応じた仕事を行なおうとしても、自身が他のジョブに邪魔されて遅れた場合には、自身の仕事に対応した気筒及びクランク位置に係る情報を知ることができなくなる。このため、クラセンジョブ及び低優先クラセンジョブ中では、多重待ち状態でないときにOS用のシステム変数S_ACASをユーザー用変数ACASとして取込み、このユーザー変数ACASをジョブ実行毎に更新して多重要求の場合にも、自身に対応した気筒及びクランク位置に係る情報を得て適正な処理がなさ

れるようにするのである。

【0100】その後、上記ステップS503あるいは上記ステップS504からステップS505へ進み、ジョブのワークエリアを設定すると、ステップS506で、レベルゼロの割込みを許可し、ステップS507で、クラセンジョブのセクションに移る。そして、このクラセンジョブセクションにリンクされた処理を実行し、ステップS508で、割込みを禁止してルーチンを抜ける。

【0101】次に、ステップS500で、これから実行すべきジョブがクラセンジョブでないときには、ラベルALJ10のステップS510で、2msジョブでないか否かを調べ、2msジョブのとき、ステップS511で、ジョブのワークエリアを設定すると、ステップS512で、レベルゼロの割込みを許可し、ステップS513で、2msジョブのセクションに移る。そして、このセクションにリンクされているジョブ本体（ユーザー側の制御ストラテジーに基づくルーチン、あるいは、OS側で用意したサービ斯拉ーチン）を実行し、ステップS514で、割込みを禁止してルーチンを抜ける。

【0102】一方、上記ステップS510で、実行すべきジョブが2msジョブでないときには、ステップS510からステップS520へ分岐し、実行すべきジョブが4msジョブか否かを調べる。そして、4msジョブでないときには、ラベルALJ30へ分岐し、4msジョブのときには、ステップS521で、ジョブのワークエリアを設定すると、ステップS522へ進む。尚、この4msジョブは、A/D変換利用ジョブであり、後述するシステムシフトバッファSSHBを介してA/D変換データを利用する。

【0103】ステップS522では、レベルゼロの割込みを許可し、次いで、ステップS523へ進むと、スイッチ入力を読み込み、ステップS524で、4msジョブのセクションに移って、リンクされているジョブ本体を実行する。その後、4msジョブのセクションから抜けると、ステップS525で、割込みを禁止し、ステップS526へ進んで、システムシフトバッファSSHBをシフトしてルーチンを抜ける。

【0104】上記システムシフトバッファSSHBは、図16に示すように、8チャンネルの各A/D変換結果がストアされる先頭オフセットアドレス0、+8、+16、+24、+32、+34、+36、+38番の各メモリ、及び、4ms毎のクランク同期のA/D変換結果がストアされる先頭オフセットアドレス-2番地の1ワードのメモリからなり、0.5ms毎に実行される1回のA/D変換結果が1ワード（2バイト）でストアされる。

【0105】先頭オフセットアドレス0番地からは、4段のシフトメモリとなっており、90°CA毎のA/D変換結果がストアされ、最新4データ（1回転分）をジョブから参照することができる。また、先頭オフセットアドレス+32、+34、+36、+38番地は、各1

ワードのメモリであり、なまし処理機能が選択されたとき、A/D変換結果を加重平均した値がストアされてノイズ除去と精度向上を図ることができるようになっており、これらのメモリのデータは、低速ジョブで利用できる。

【0106】また、各先頭オフセットアドレス+8、+16、+24番地からは、各4ワードのメモリであり、4msジョブで利用できるようになっている。これらの各メモリは、先頭に最新のA/D変換結果がストアされ、4msジョブのオーバーラップカウンタOLCの値に応じて順にシフトされ、先にストアしたデータから読出されるFIFOバッファとなっている。

【0107】すなわち、A/D変換は、0.5ms毎の定期割込みにより4ms周期で正確に行なわれるが、4msジョブは優先度の高いジョブに邪魔されて遅れることがある。従って、A/D変換の受渡しにFIFOバッファを用い、4msジョブで+8、+16、+24番地の各FIFOバッファのデータを参照後、上記ステップS526で、各FIFOバッファのデータを順にシフトするのである。

【0108】一方、上記ステップS520で、実行すべきジョブが4msジョブでなく、ラベルALJ30へ分岐したときには、ステップS530で、実行すべきジョブが10msジョブか否かを調べ、10msジョブのとき、ステップS531で、ジョブのワークエリアを設定し、ステップS532で、レベルゼロの割込みを許可すると、ステップS533で、10msジョブのセクションに移って、ジョブ本体を実行し、ステップS534で割込みを禁止してルーチンを抜ける。

【0109】尚、上記10msジョブのセクションには、半回転時間からエンジン回転数を算出するサービ斯拉ーチン、前述した点火スケジュールを作成するサービ斯拉ーチンなどがOS側で用意されている。

【0110】また、上記ステップS530で、実行すべきジョブが10msジョブでないときには、上記ステップS530からステップS540へ分岐し、実行すべきジョブが低優先クラセンジョブであるか否かを調べる。そして、低優先クラセンジョブでないときには、上記ステップS540からラベルALJ50へ分岐し、実行すべきジョブが低優先クラセンジョブのときは、上記ステップS540からステップS541へ進んで、現在の状態が多重待ち状態であるか否かを調べる。

【0111】そして、現在の状態が多重待ち状態でないときには、上記ステップS541からステップS542へ進んで、システム変数S_ACAS（クランク総合位置変数）をユーザー変数ACASとしてステップS544へ進み、多重待ち状態のときには、上記ステップS541からステップS543へ分岐し、ユーザー変数ACASを一つ増やして12で割った剰余をとった後、ステップS544へ進む。

【0112】ステップS544では、ジョブのワークエリアを設定し、ステップS545で、レベルゼロの割込みを許可すると、ステップS546で、低優先クラセンジョブのセクションに移り、ジョブ本体を実行した後、ステップS547で割込みを禁止し、ルーチンを抜ける。

【0113】さらに、ラベルALJ50では、先ずステップS550で、実行すべきジョブが50msジョブであるか否かを調べ、50msジョブのときには、ステップS551へ進んでジョブのワークエリアを設定し、ステップS552へ進む。

【0114】ステップS552では、レベルゼロの割込みを許可すると、ステップS553で、50msジョブのセクションに移り、OS側で用意したエンスト処理ルーチン、気筒別の点火時期リタードルーチン、燃料噴射開始時期設定ルーチンなどを実行し、また、ユーザ側の制御ストラテジーに基づくルーチンを実行する。そして、ジョブの終了後、ステップS554で割込みを禁止し、ルーチンを抜ける。

【0115】一方、上記ステップS550で実行すべきジョブが50msジョブではないときには、上記ステップS550からステップS560へ分岐し、ジョブのワークエリアを設定すると、ステップS561で、レベルゼロの割込みを許可し、ステップS562へ進んで、250msジョブのセクション領域へ移行し、ジョブ本体を実行後、ステップS563で割込みを禁止してルーチンを抜ける。

【0116】以上のジョブ優先処理においては、クランク位置を常的に確に把握しておく必要があり、クラセン割込み毎に、図9に示すクランク位置算出サブルーチンが実行されて前述したクランク位置変数S_{CCAS}、S_{RCAS}、S_{ACAS}、S_{ECAS}が算出される。尚、以下の説明においては、システム変数であることを示すSを省略する。

【0117】このクランク位置算出サブルーチンでは、まず、ステップS600で、クラセンタイマの下2バイトをソフトタイマの下2バイトにストアする。このクラセンタイマはECU50に備えられたハードウェアタイマであり、本実施例においては、16ビットタイマで最大255msまで計数が可能であるが、メモリ上に3バイトの連続した領域を確保して下2バイトにクラセンタイマの2バイトを転写し、クラセンタイマのオーバーフローにより発生する割込みで3バイト目をカウントアップすることによりソフトタイマとして使用する。

【0118】すなわち、クラセンタイマにオーバーフローが発生したとき、図11のオーバーフロー割込みが発生し、ステップS900で、ソフトタイマの3バイト目を1増やすと、ステップS901で、ソフトタイマの3バイト目が255を越えていないかを判別する。そして、255を越えていないときには、ステップS901からステップS903へジャンプし、255を越えているときには、ステップS901からステップS902へ進んでソフトタイマの3バ

イト目を255で止め、ステップS903へ進む。

【0119】ステップS903では、ソフトタイマの3バイト目が2でないか否か、すなわち、クラセン間隔が0.5sec(255ms×2)を越えてエンスト状態であるか否かを調べる。そして、ソフトタイマの3バイト目が2でないときには割込みを終了し、ソフトタイマの3バイト目が2のときには、上記ステップS903からステップS904へ進んで、エンストフラグが立っているか否かを調べる。

10 【0120】上記ステップS904では、エンストフラグが立っているときには割込みを終了し、エンストフラグが立っていないときには、ステップS905で、前述した50msジョブに用意されているエンスト処理ルーチンを起動させるエンスト処理要求フラグを立てて割込みを終了する。

【0121】これにより、クラセン間隔を最大64sec(255ms×256)まで計数することが可能となり、16ビット以上の特別なハードウェアタイマを使用することなく、クランキング時などクラセン間隔が極めて長い場合にも容易に対応することができる。

20 【0122】一方、クランク位置算出のサブルーチンでは、ステップS600からステップS601へ進むと、クラセン間隔が設定時間以下か否かを調べる。この設定時間は、最大エンジン回転数に対応するクラセン間隔としての時間、例えば0.3msであり、上記ステップS601でクラセン間隔が設定時間以下のときには、ノイズの混入などによるクラセンタイマの計数エラーとしてステップS602でアキュムレータAにエラーコード1を格納し、上記ステップS601で、クラセン間隔が設定時間より長いときには、クラセンタイマの計数が正常であるとしてステップS603へ進む。

30 【0123】ステップS603では、後述するCCAS・RCAS判別サブルーチンを実行してクランク位置を判別し、ステップS604で、エラーレベルECASが2であるか否か、すなわち、クランキング時などのように気筒判別がなされていない状態であるか否かを調べ、ECAS=2のときには、ステップS605へ分岐してアキュムレータAにエラーコード1を格納し、ルーチンを抜ける。

40 【0124】一方、上記ステップS604で、ECAS≠2のときには、ステップS606へ進み、ソフトタイマの3バイト目を0とする。そして、ステップS607へ進み、クランク位置情報変数CCASが1であるか否か、すなわち、現在のクランク位置がBTDC65°CA~10°CAの間(図14参照)であるか否かを調べ、CCAS=1のときには、ステップS607からステップS609へジャンプし、CCAS≠1のときには、ステップS607からステップS608へ進んで、A/D変換リクエストを1増加させ、ステップS609へ進む。

50 【0125】このA/D変換リクエストは、クランク角90°毎にクランク同期A/D変換を指示するためのフ

ラグ的な変数であり、0、1の値をとり、値が1のときクランク同期A/D変換を指示する。すなわち、前述したように、8チャンネルのA/D変換のうち1チャンネルのA/D変換はクランク角90°毎に行なわれるが、CCASが0になったとき(BTDC97°)と、CCASが2になったとき(BTDC10°)、クランク同期のA/D変換リクエストをセットし、0.5ms毎のA/D変換順番に対してクランク角90°毎のA/D変換を割込ませるのである。

【0126】その後、ステップS609では、気筒情報変数RCASを3倍してクランク位置情報変数CCASを加算することにより総合位置変数ACASを算出すると(ACAS=RCAS×3+CCAS)、ステップS610で、ソフトタイマを2バイトでリミットし、クラセンタイマがオーバーフローしている場合には下2バイトをFFFF(255ms)としてステップS611へ進む。

【0127】ステップS611では、総合位置変数ACAS=0, 1, 2, ..., 11を添字とする配列TCAS[ACAS](配列の要素)にクラセン間隔データをストアし、ステップS612で、クランク位置情報変数CCAS=0, 1, 2を添字とする配列MTCSX[CCAS](配列の要素)にクラセン間隔データをストアする。

【0128】配列TCASは、図17(a)に示すように、ACAS=0, 1, 2, ..., 11に対応するエンジン2回転分のクラセン間隔データがストアされた12ワードのクラセン間隔テーブルであり、配列MTCSXは、図17(b)に示すように、CCAS=0, 1, 2に対応する3つのクラセン間隔データがストアされた3ワードのクラセン間隔テーブルである。

【0129】すなわち、配列TCASを参照することにより、各気筒の各クランク位置におけるクラセン間隔の変化(回転速度の変化)を知ることができ、各気筒の失火の有無、燃焼状態などを判断することができ、全気筒の運転状況を把握することができる。また、配列MTCSXを参照することにより、常に最新のクラセン間隔を得ることができ、現在の運転状況を迅速に把握することができる。

【0130】次いで、ステップS613へ進むと、再び、エラーレベルECASの値を調べる。ここでは、前述のステップS604においてECAS≠2であることを既に確認してあるため、エラーレベルECASが1か否か、すなわち、クランク位置の判別が不安の残る推定状態であるか否かを調べる。

【0131】上記ステップS613でECAS≠1(すなわちECAS=0)であり、クランク位置が確証をもって判別されているときには、上記ステップS613からステップS614へ進んで、最新3つのクラセン間隔データの和(配列MTCSXの和)を、3バイトの半回転時間MTCS18として算出する(MTCS18=ΣMTCSX)。すなわち、半回転時間MTCS18は、クラセン

割込み毎にクランク位置情報変数CCASが更新されて配列MTCSX内のデータが更新される毎に算出され、BTDC97°, 65°, 10°の各位置毎にクラセン間隔の移動和を取ることにより常に最新のデータが得られるようになっている。

【0132】一方、上記ステップS613で、ECAS=1であり、クランク位置の判別が不安の残る推定状態であるときには、配列MTCSXから半回転時間MTCS18を算出せず、ステップS615で半回転時間推定のサブルーチンを実行して半回転時間MTCS18を推定する。

【0133】この半回転時間MTCS18の推定について説明すると、まず、図10のステップS700で、クランク位置情報変数CCASが1か否か、すなわち、現在のクランク位置がBTDC65°~10°の間か否かを調べ、CCAS=1のときには、ステップS701で、BTDC97°~65°間の角度32°から、前回のクラセン間隔×180/32を半回転時間MTCSXと推定してリターンする。

【0134】一方、上記ステップS700で、CCAS≠1のときには、上記ステップS700からステップS702へ進んで、クランク位置情報変数CCASが2か否か、すなわち、現在のクランク位置がBTDC10°からATDC83°(次の気筒のBTDC97°)の間か否かを調べ、CCAS≠2のとき(CCAS=0のとき)には、ステップS703へ進んで、BTDC10°~ATDC83°間の角度93°から前回のクラセン間隔×180/93を半回転時間MTCSXと推定してリターンする。また、上記ステップS702で、CCAS=2のときには、上記ステップS702からステップS704へ分岐し、BTDC65°~10°間の角度55°から前回のクラセン間隔×180/55を半回転時間MTCSXと推定してリターンする。

【0135】そして、クランク位置算出サブルーチンでは、ステップS614で半回転時間MTCS18を算出した後、あるいは、以上説明したステップS615のサブルーチンにより半回転時間MTCS18を推定した後は、ステップS616へ進み、3バイトの半回転時間MTCS18を2バイトにリミットして所定の変数MTCSKにストアすると、ステップS617で、この変数MTCSKを2倍して変数MTCSK4にストアし、ステップS618で、正常終了コード0をアキュムレータAに格納してルーチンを抜ける。

【0136】次に、図1に示されるCCAS・RCAS判別サブルーチンについて説明する。このサブルーチンでは、まず最初に、ステップS800で、カムセンカウンタを0~4にリミットする。このカムセンカウンタで計数されるカム角センサ41からのカムパルスの数は、図19に示すように、正常状態の場合0~3であるが、ノイズなどの影響により4以上の異常な計数値となるおそれがあるため、カムセンカウンタを0~4にリミットして

異常な状態を4で代表するのである。

【0137】次に、ステップS801へ進み、カムセンカウンタ（の計数値）、気筒情報変数RCAS、クランク位置情報変数CCASから、 $5 \times 4 \times 2$ の組合わせ（カムセンカウンタが0～4の5種類、気筒情報変数RCASが0～3の4種類、クランク位置情報変数CCASが0、1と2の場合の2種類）に対する状態データがストアされている気筒・クランク位置状態マップCCHMAPを読む。

【0138】この気筒・クランク位置状態マップCCHMAPは、図18（a）及び（b）に示すように、クランク位置情報変数CCASが0あるいは1の場合と、気筒情報変数RCASの変化点であるクランク位置情報変数CCASが2の場合とに分け、カムセンカウンタと気筒情報変数RCASの各組合わせの起こり得る全ての状態に対し、正常か異常か、確定して良いか推定すべきかを示す状態データがストアされている。

【0139】この状態データは2ビットのデータであり、ビット0の値により確定か推定かを表わし、ビット1の値により正常か異常かを表わす。ビット0の値は、0のとき確定、1のとき推定を示し、図18からわかるように、カムセンカウンタが2、3の場合にのみ確定であって、それ以外は推定せざるを得ない状態である。また、ビット1の値は、0のとき正常、1のとき異常を示し、カムセンカウンタが3以下で、且つ、図14及び図19による組合せに合致する場合のみ正常であって、それ以外は異常な状態である。

【0140】例えば、CCAS=0あるいは1、すなわち、ある気筒のBTDC97°～10°に対し、カムセンカウンタが0で気筒情報変数RCASが0となる組合せは、図14及び図19からもわかるように、クランク位置を正常に推定すれば良い状態であるため、気筒・クランク位置状態マップCCHMAPの該当領域に2進数で01（正常推定）の状態データがストアされており、さらに、カムセンカウンタが1で気筒情報変数RCASが0となる組合せは、明らかに異常であって推定するしかない状態であるため、気筒・クランク位置状態マップCCHMAPの該当領域に2進数で11（異常推定）の状態データがストアされている。

【0141】また、CCAS=2、すなわち、ある気筒のBTDC10°～ATDC83°に対し、カムセンカウンタが3で気筒情報変数RCASが0となる組合せは、#1気筒のTDCを挟んだクランク位置と正常に確定できるため、気筒・クランク位置状態マップCCHMAPの該当領域に2進数で00（正常確定）の状態データがストアされており、さらに、カムセンカウンタが2で気筒情報変数RCASが0となる組合せは、明らかに異常ではあるがカムセン入力に2ヶある以上確定せざるを得ない状態であるため、気筒・クランク位置状態マップCCHMAPの該当領域に2進数で10（異常確定）

の状態データがストアされている。

【0142】そして、上記ステップS801で気筒・クランク位置状態マップCCHMAPから状態データを読み込むと、ステップS802へ進み、エラーレベルECASが2でないか否か、すなわち、現在の状態が気筒判別のなされていない不明な状態であるか否かを調べ、ECAS=2のときには、ステップS803で気筒・クランク位置状態マップCCHMAPから読み込んだ状態データのビット0が0か否か、すなわち確定状態か否かを調べ、確定状態のときにはステップS804へ進み、確定状態でなく推定状態であるときにはルーチンを抜けて確定状態となるまで待つ。

【0143】一方、上記ステップS802でECAS≠2のときにはステップS804へ進んで、推定状態か否かを調べ、確定状態、推定状態に応じてステップS805以降の処理あるいはステップS812以降の処理へ進む。また、上記ステップS803において確定状態でステップS804へ進んだときには、ステップS805以降の処理へ進む。

【0144】まず、ステップS805以降の処理について説明すると、このステップS805へ進んだときには、正常、異常に拘らず気筒判別がなされた確定状態であるため、図19のタイムチャートからもわかるように、今回のクラセン割込みはカムパルスが3ヶあるいは2ヶ入力された後のBTDC97°の割込みであるため、クランク位置情報変数CCASを0にする。

【0145】次いで、ステップS806へ進んでカムセンカウンタが3でないか否かを調べ、カムセンカウンタが3でないとき、すなわちカムセンカウンタが2のときには、#2気筒の点火後であるため、ステップS807で気筒情報変数RCASを3にしてステップS809へ進み、カムセンカウンタが3のときには、#1気筒の点火後であるため、ステップS808で気筒情報変数RCASを1にしてステップS809へ進む。

【0146】ステップS809では、更新したクランク位置情報変数CCAS、気筒情報変数RCAS、及び、カムセンカウンタをパラメータとして再び気筒・クランク位置状態マップCCHMAPから状態データを読み込み、この状態データのビット1が1であるか、すなわち異常状態であるか否かを調べる。

【0147】その結果、上記ステップS809において、状態データのビット1が1で異常状態と判定されるときには、クランク位置情報変数CCAS、気筒情報変数RCASの更新結果は不安の残る推定であるとしてステップS810でエラーレベルECASを1にしてルーチンを抜け、状態データのビット1が0であり正常状態であるときには、ステップS811でエラーレベルECASを0としてルーチンを抜ける。

【0148】一方、ステップS812以降の処理では、ステップS812で、現在のクランク位置情報変数CCAS（前回のクラセン割込みで算出されたクランク位置情報変数

CCAS)が2、すなわち、気筒情報変数RCASの変化点であるか否かを調べ、CCAS=2のときには、ステップS812からステップS815へ進んで、気筒情報変数RCASを1増加させ、ステップS816でクランク位置情報変数CCAS=0にしてステップS817へ進む。

【0149】一方、上記ステップS812で、CCAS≠2のときには、上記ステップS812からステップS813へ進んでカムセンカウンタが0でないか否かを調べ、カムセンカウンタが0でないときには、前述のステップS815へ分岐し、カムセンカウンタが0のときには、ステップS814でクランク位置情報変数CCASを1増加させ、ステップS817へ進む。

【0150】ステップS817では、更新したクランク位置情報変数CCAS、気筒情報変数RCAS、及び、カムセンカウンタをパラメータとして再び気筒・クランク位置状態マップCCHMAPから状態データを読み込んで異常状態であるか否かを調べ、状態データのビット1が0であり、正常状態であるときには、そのまま(現在のエラーレベルECAS=0のまま)ルーチンを抜ける。また、上記ステップS817で、状態データのビット1が1で異常状態のときには、クランク位置情報変数CCAS、気筒情報変数RCASの更新結果は不安の残る推定であるとしてエラーレベルECASを1にし、ルーチンを抜ける。

【0151】以上のように、クランク位置変数CCAS、RCAS、ACAS、ECASにより刻々と変化するクランク位置の状態を把握し、さらに、気筒・クランク位置状態マップCCHMAPを用いて、現在の状態を評価し、次にとるべき状態を知ることができるため、通常の条件判断による処理に比較して大幅にプログラムを簡略化することができ、メモリ容量の節約、処理速度の向上を図ることができるとともに、プログラムの可携性を向上して仕様変更に対しても柔軟に対処することができる。

【0152】

【発明の効果】以上説明したように本発明によれば、クランク位置判別のためのプログラムを簡略化してメモリ容量の増大や処理速度の低下を抑え、刻々と変化するエンジンのクランク位置を容易に把握することができるため、メモリ容量の節約、処理速度の向上を図ることができるとともに、プログラムの可携性を向上して仕様変更に対しても柔軟に対処することができるなど優れた効果が得られる。

*

*【図面の簡単な説明】

【図1】CCAS・RCAS判別サブルーチンのフローチャート

【図2】0.5ms毎の定期割込み処理のフローチャート

【図3】クラセン割込み処理のフローチャート

【図4】ジョブ優先処理のフローチャート

【図5】ジョブ実行サブルーチンの部分フローチャート1

【図6】ジョブ実行サブルーチンの部分フローチャート2

【図7】ジョブ実行サブルーチンの部分フローチャート3

【図8】ジョブ実行サブルーチンの部分フローチャート4

【図9】クランク位置算出サブルーチンのフローチャート

【図10】半回転時間推定サブルーチンのフローチャート

【図11】クラセンタイマのオーバーフロー割込み処理のフローチャート

【図12】ジョブの実行状態を示す説明図

【図13】ジョブフラグの説明図

【図14】クランク位置変数の説明図

【図15】ジョブ実行中フラグとオーバーラップカウンタの変化を示す説明図

【図16】システムシフトバッファの説明図

【図17】クラセン間隔テーブルの説明図

【図18】気筒・クランク位置状態マップの説明図

【図19】クランク位置とエンジンの行程を示すタイムチャート

【図20】エンジン系の概略構成図

【図21】クランクロータとクランク角センサの正面図

【図22】カムロータとカム角センサの正面図

【図23】電子制御系の回路構成図

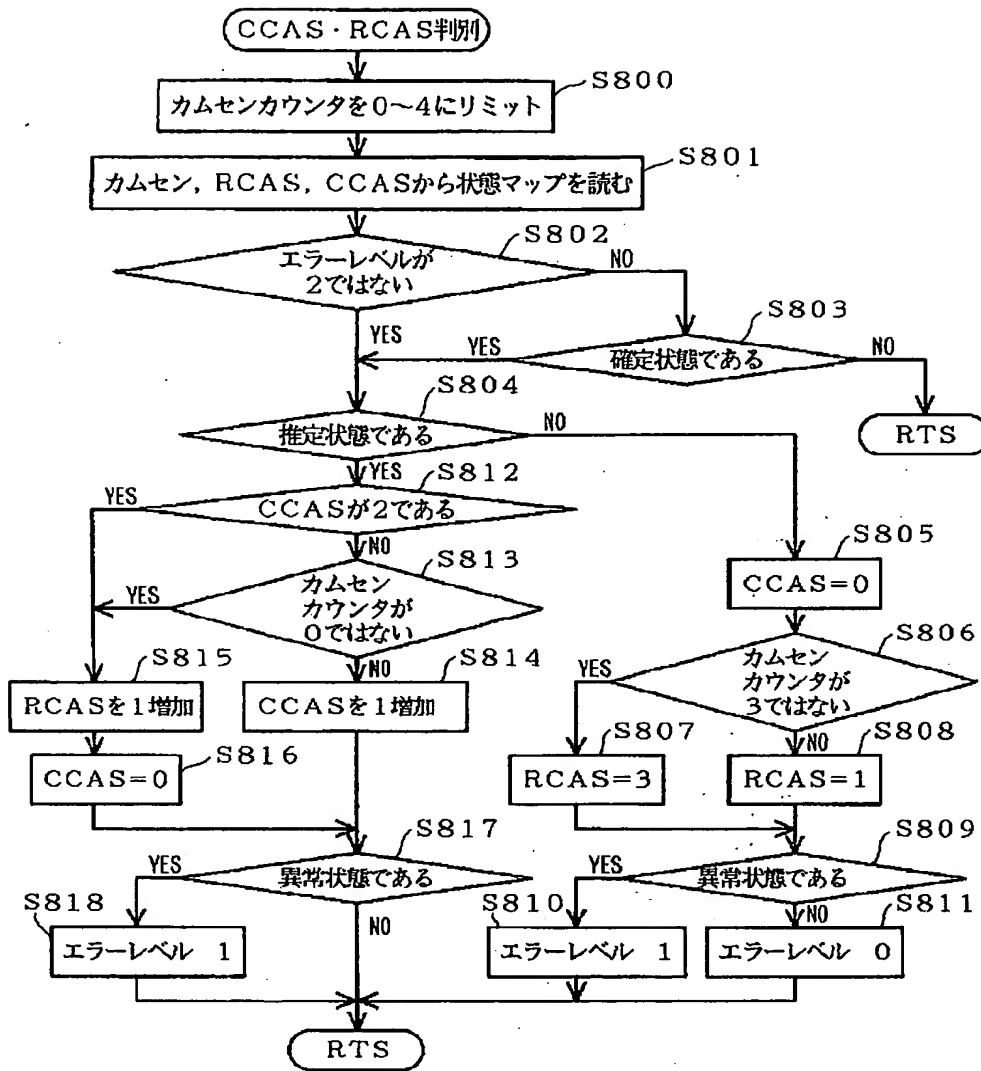
【符号の説明】

50 ECU
CCAS クランク位置情報変数
RCAS 気筒情報変数
40 ACAS 総合位置情報変数
ECAS エラーレベル
CCHMAP 気筒・クランク位置状態マップ

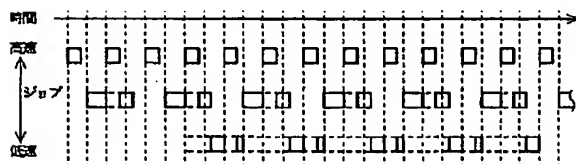
【図13】

優先レベル	7	6	5	4	3	2	1
ジョブ	クラセン	2ms	4ms	10ms	低優先クラセン	50ms	250ms

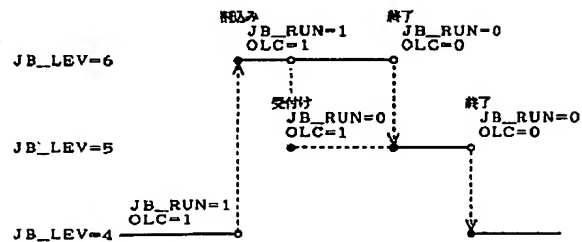
【図1】



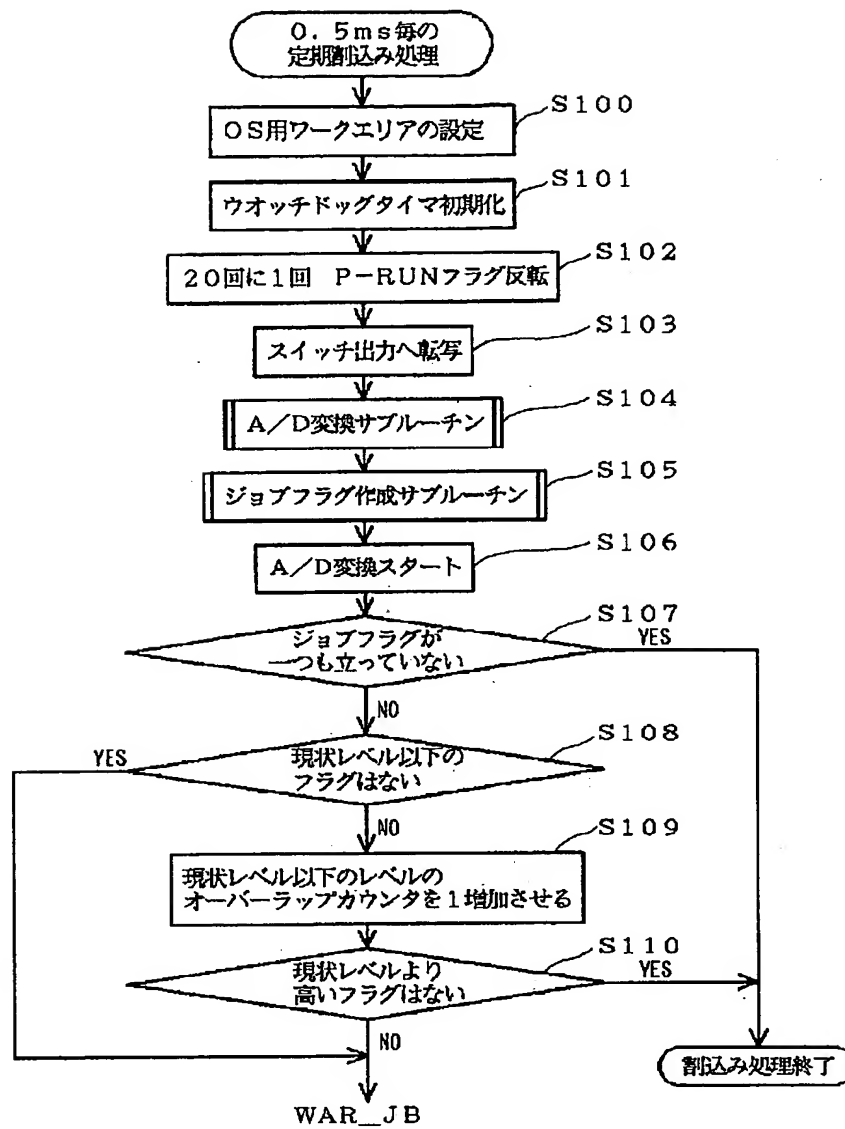
【図12】



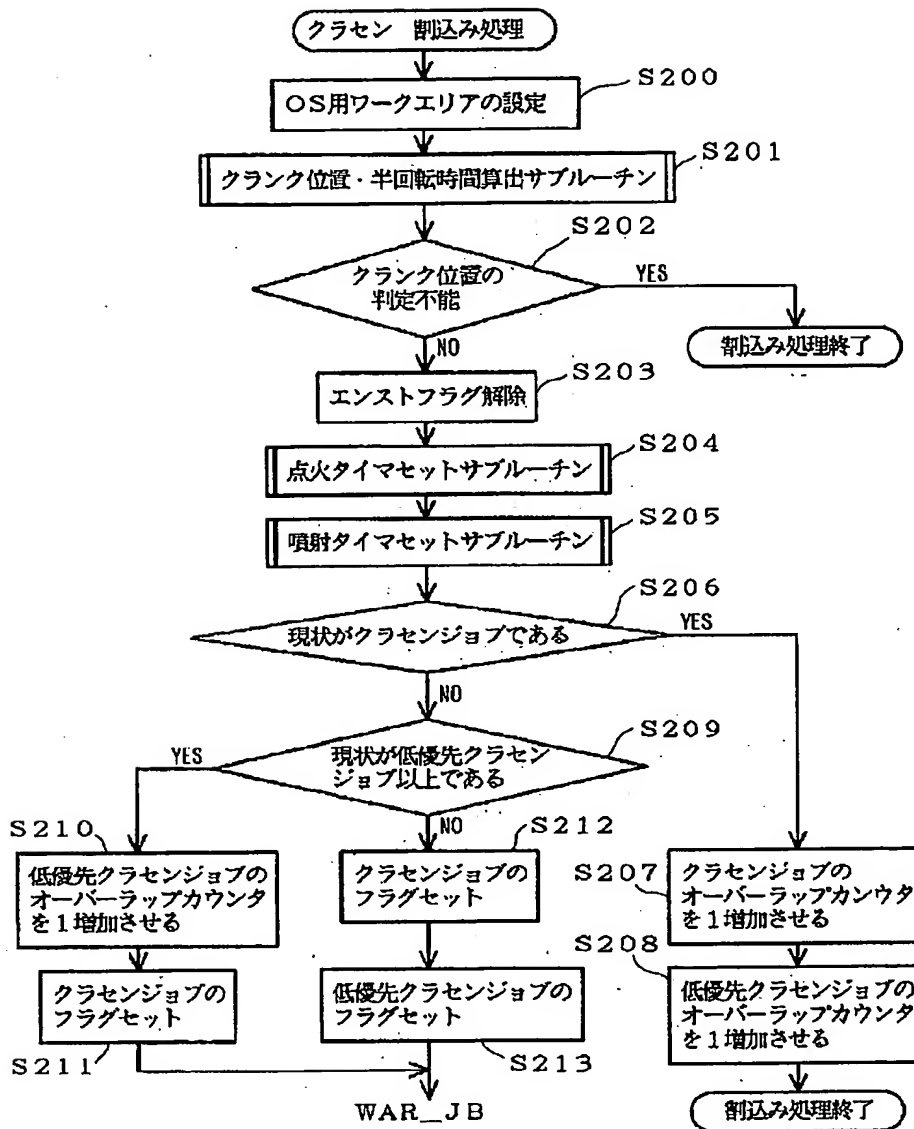
【図15】



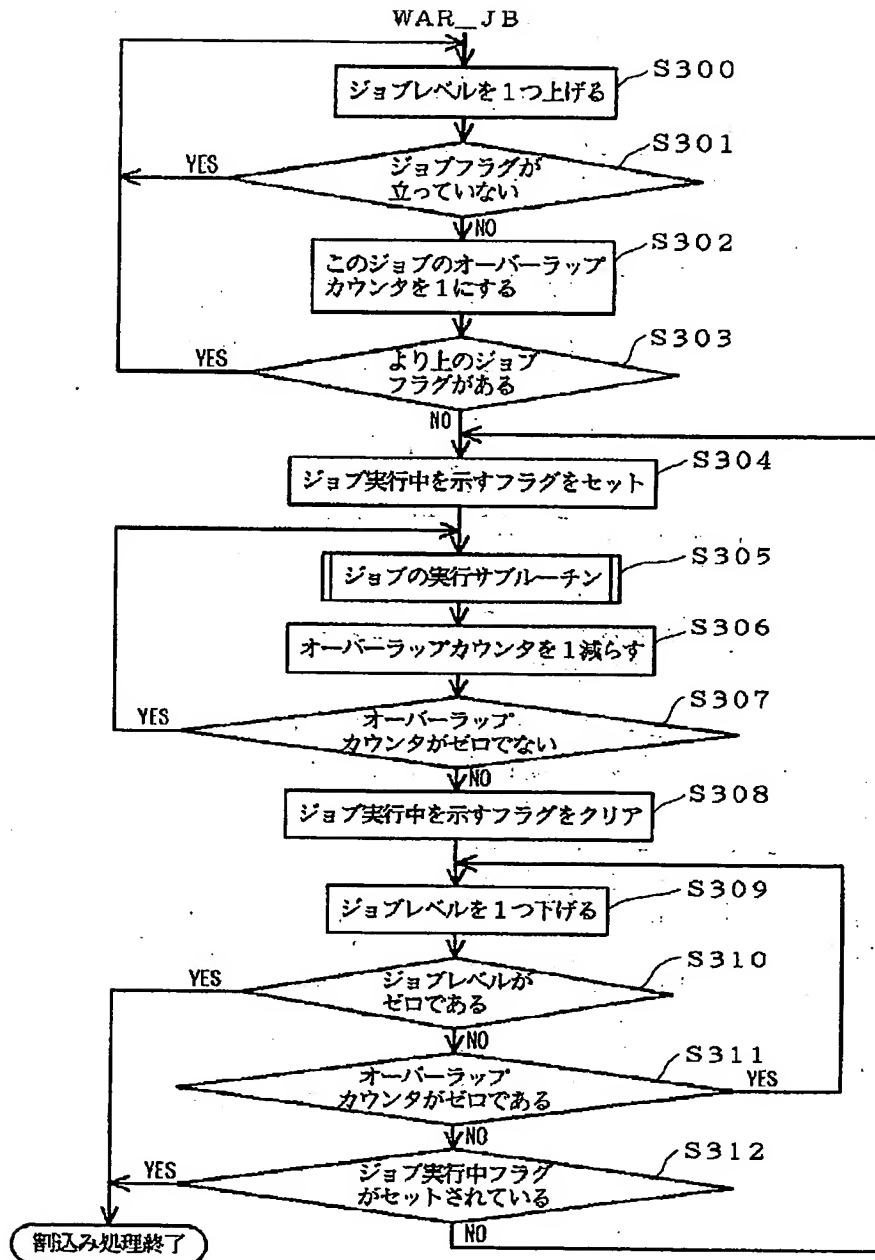
【図2】



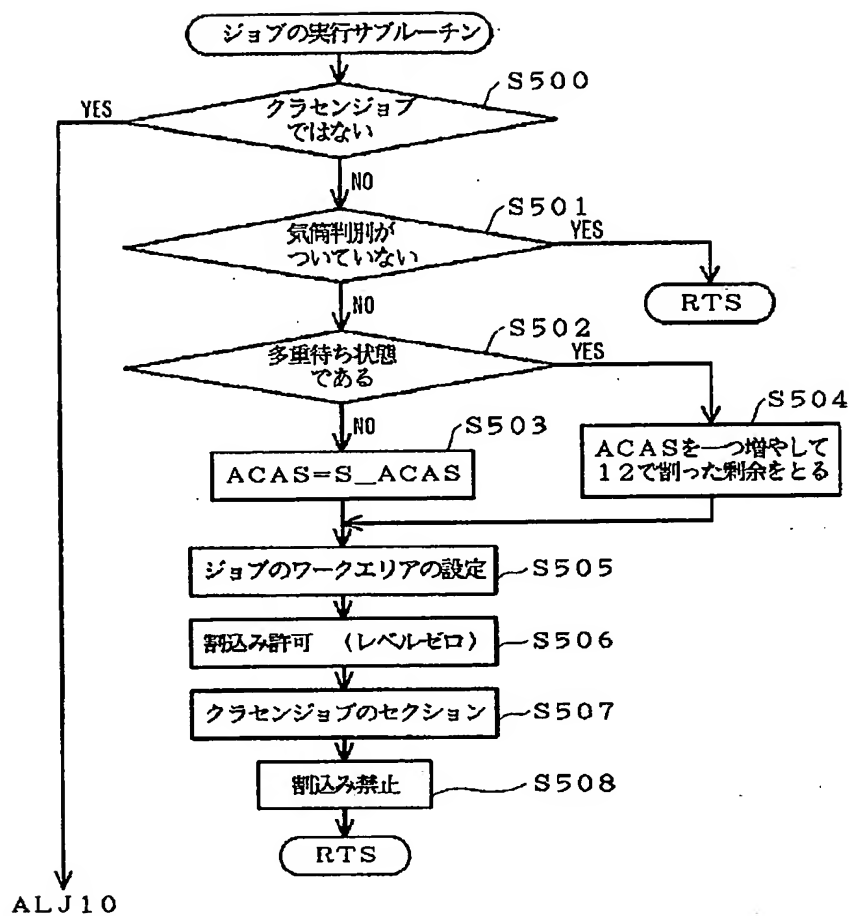
【図3】



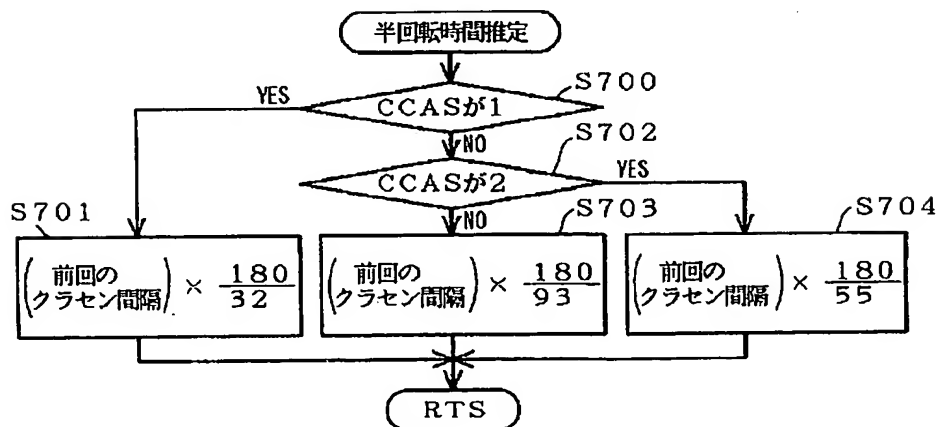
【図4】



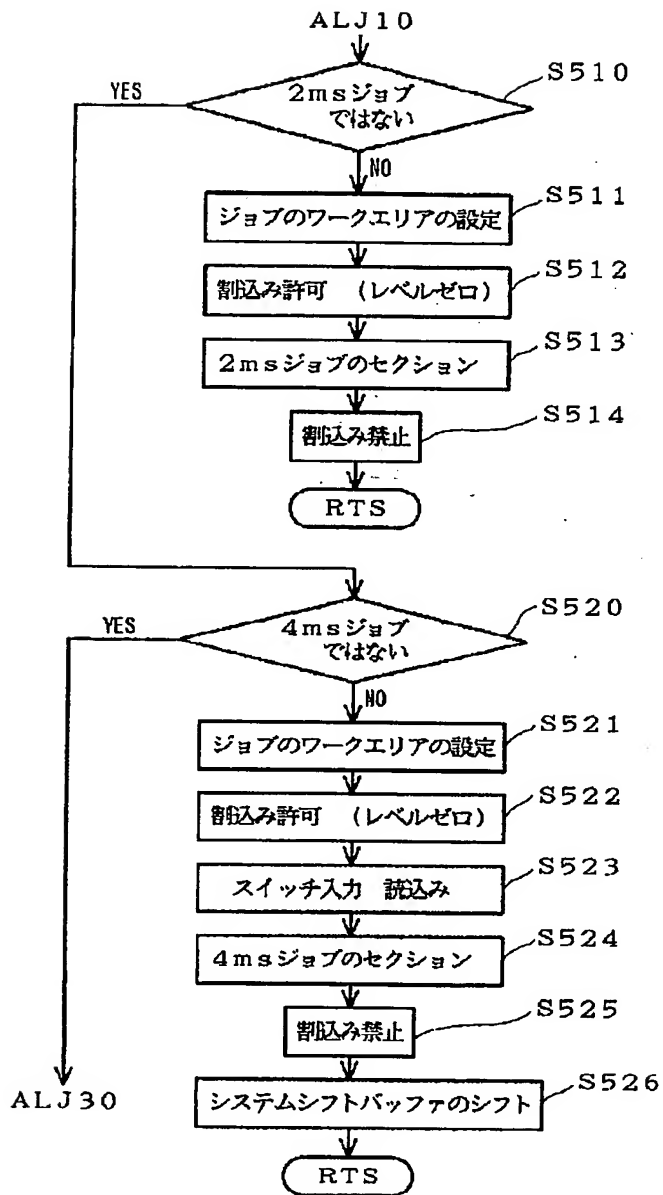
【図5】



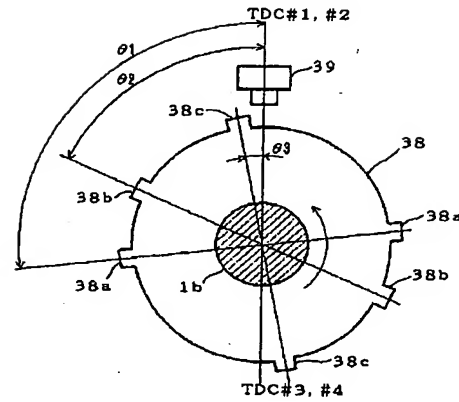
【図10】



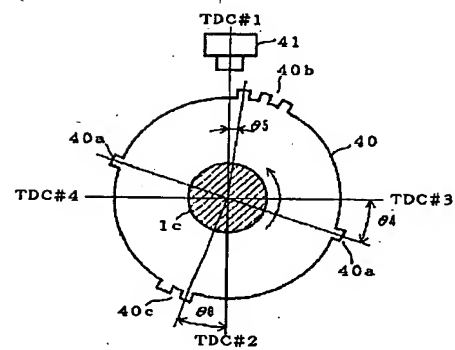
【図6】



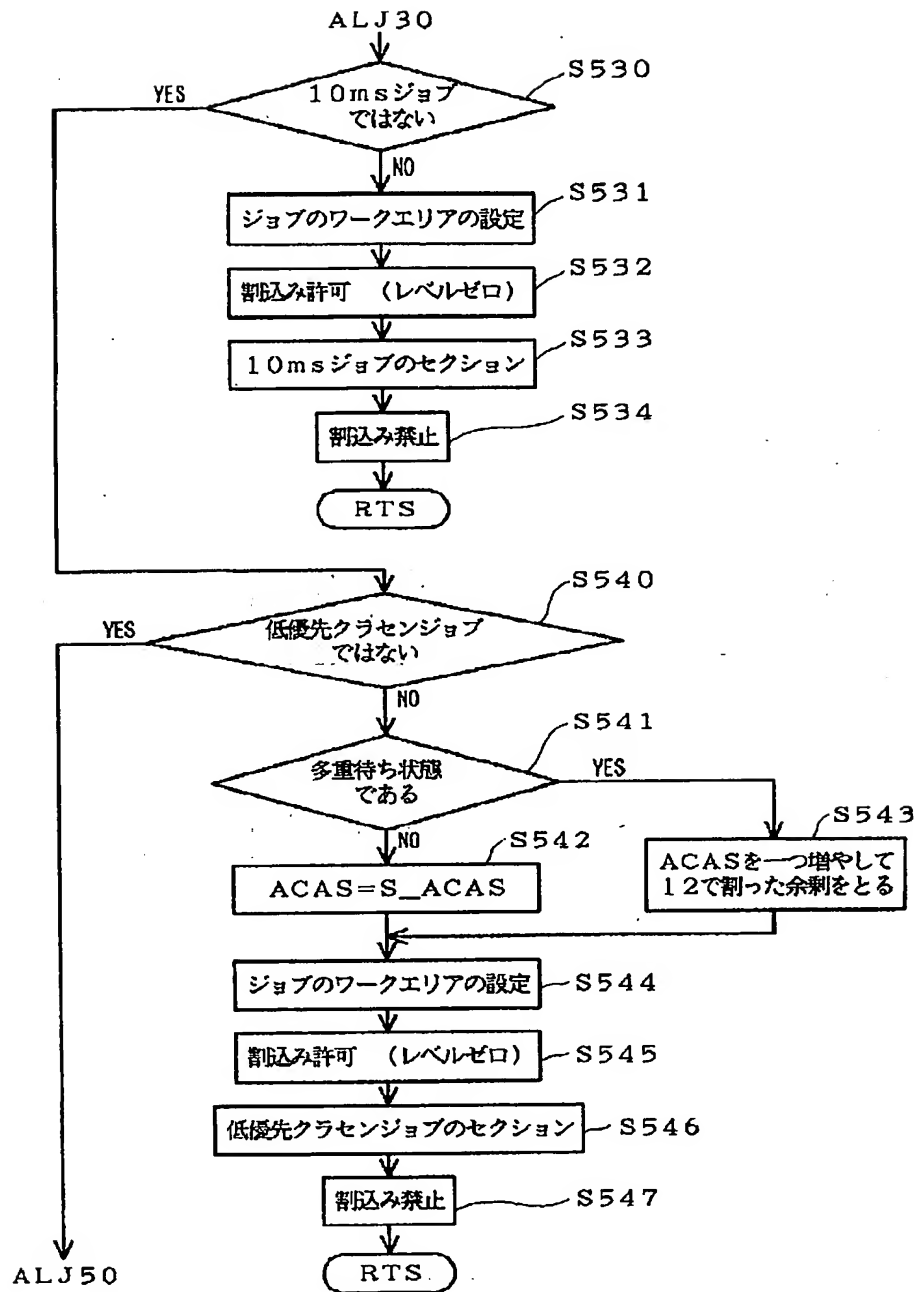
【図21】



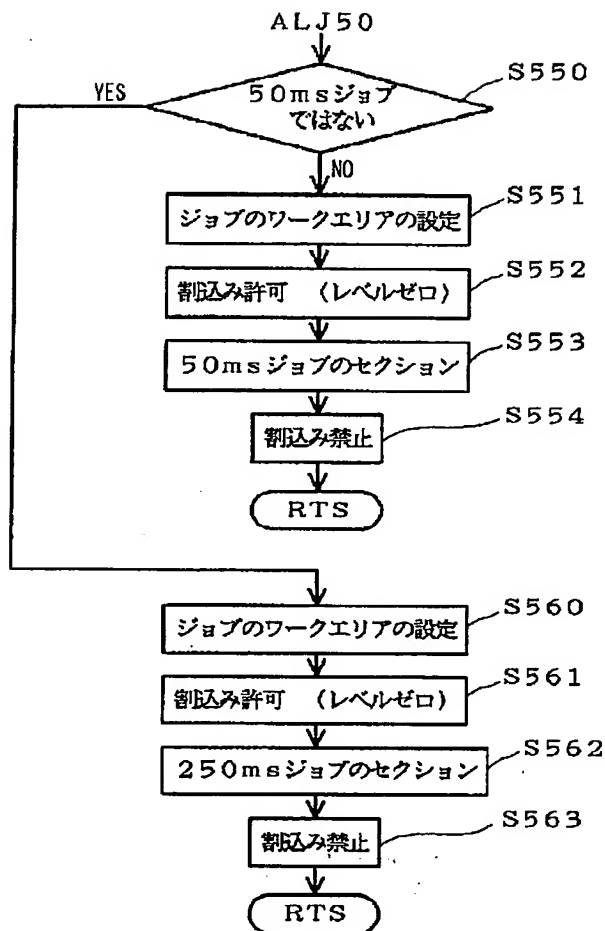
【図22】



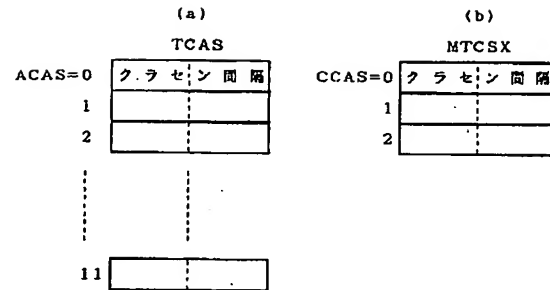
【図7】



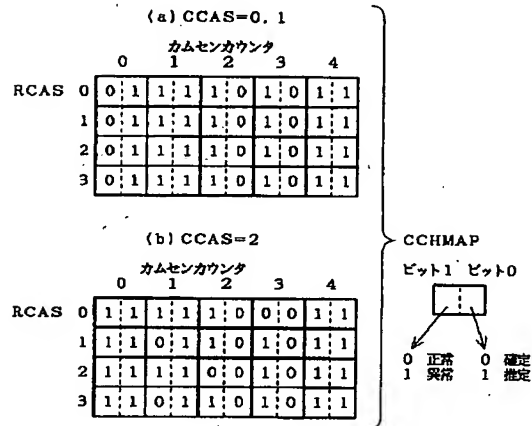
【図8】



【図17】



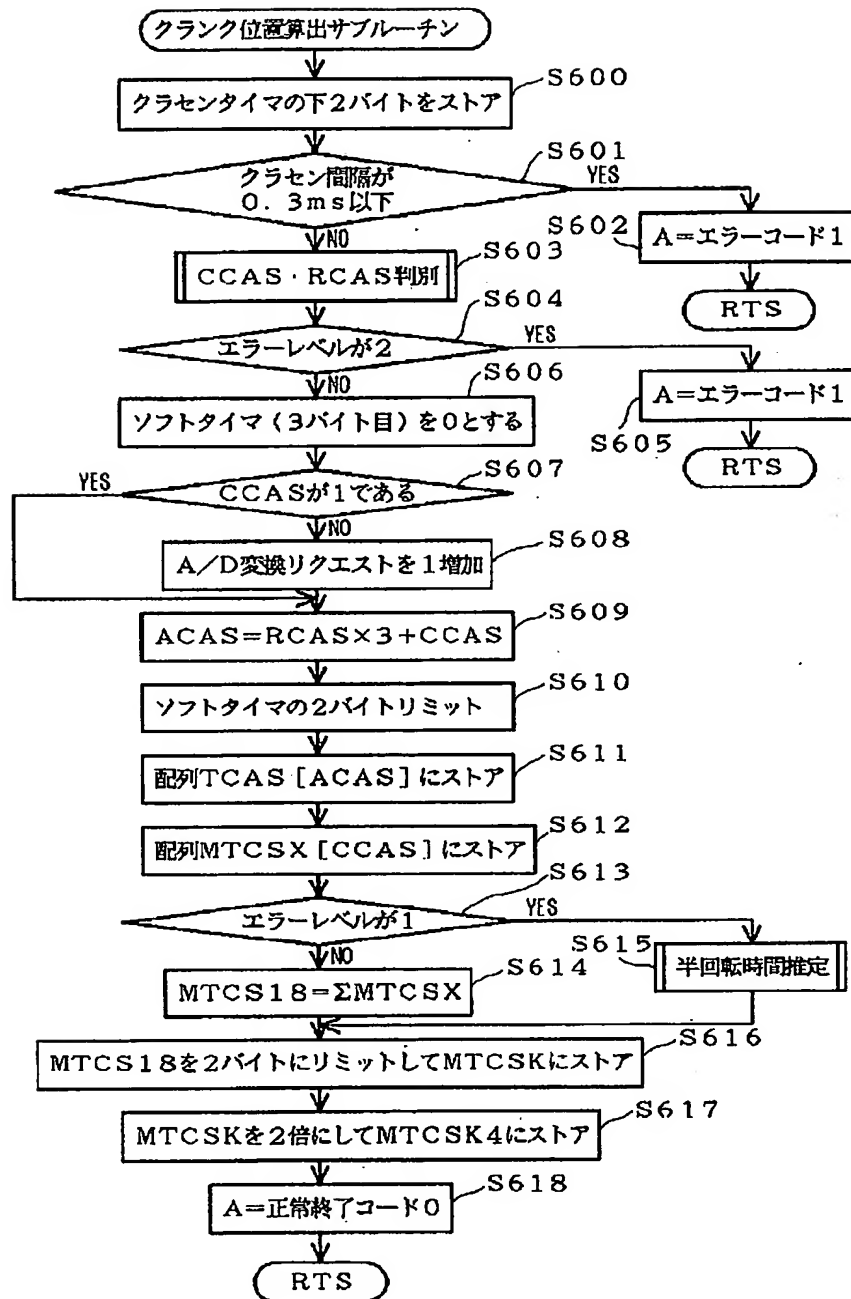
【図18】



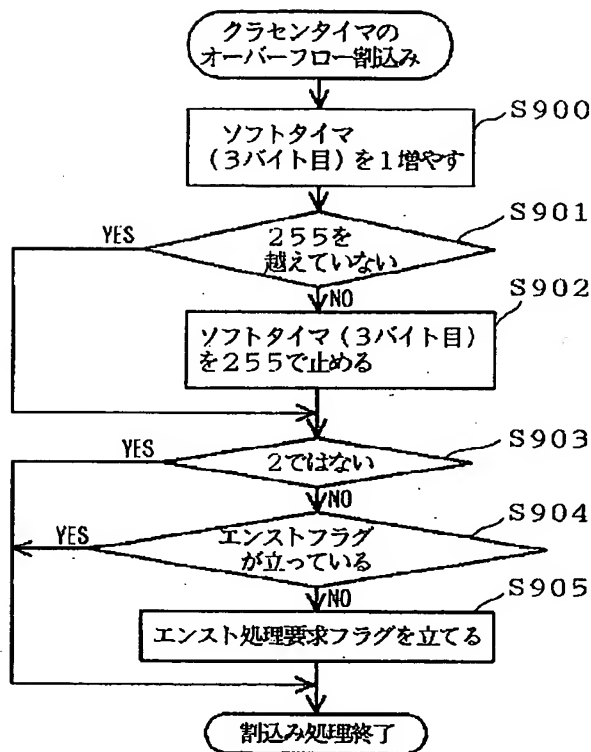
【図14】

TDC	#1				#3				#2				#4			
クラセン	97°	65°	10°	97°	65°	10°	97°	65°	10°	97°	65°	10°	97°	65°	10°	97°
CCAS	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0
RCAS	0	0	0	1	1	1	2	2	2	3	3	3	0	0	0	1
ACAS	0	1	2	3	4	5	6	7	8	9	10	11	0	1	2	3
ECAS	0: 確証を持った状態				1: 不安の残る推定状態				2: 不明な状態							

〔図9〕



【図11】



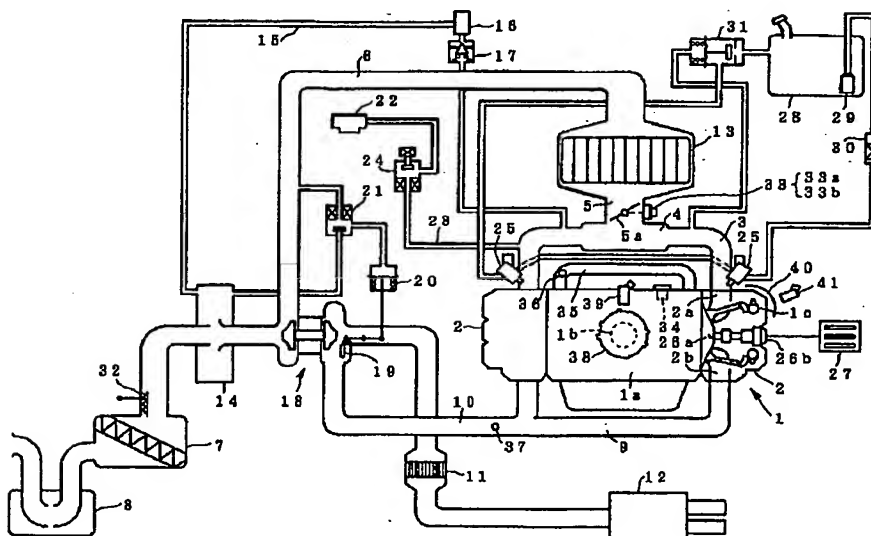
【図16】

-2					クランク同期のA/D変換結果	
0						90°毎 のA/D 変換結果
+8					経過	90°前
+16						180°前
+24						270°前
+32						
+34						
+36						
+38						

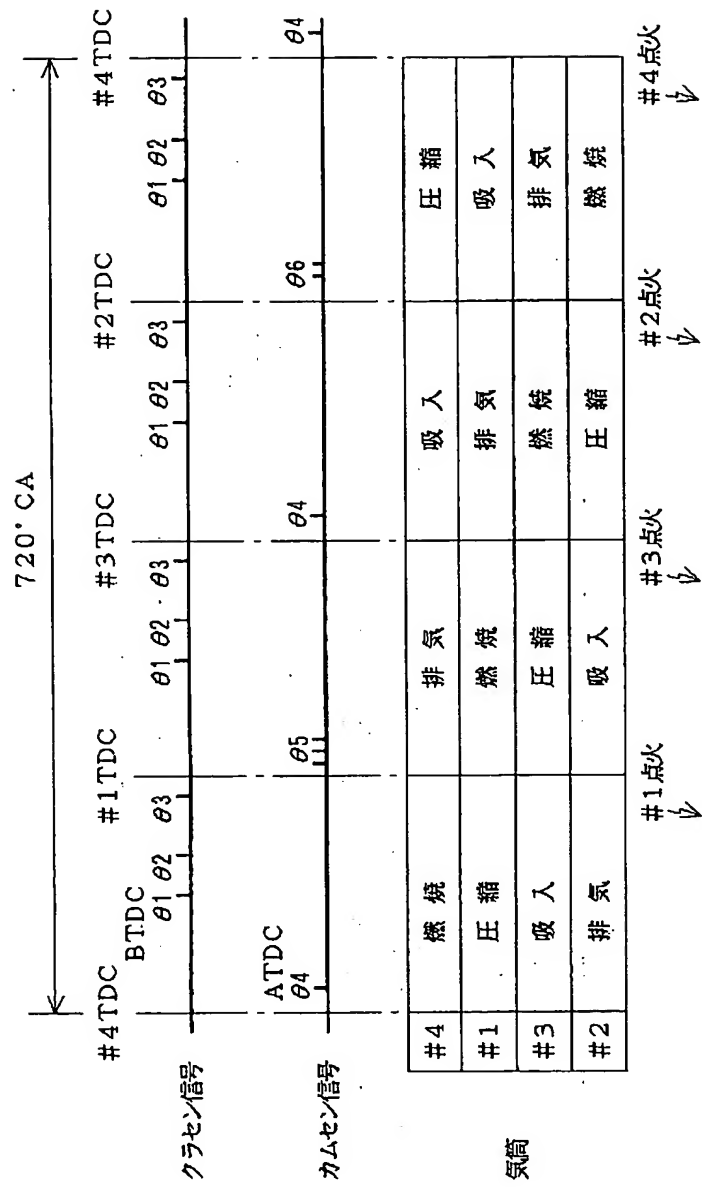
4ms
ジョブ用

低速
ジョブ用

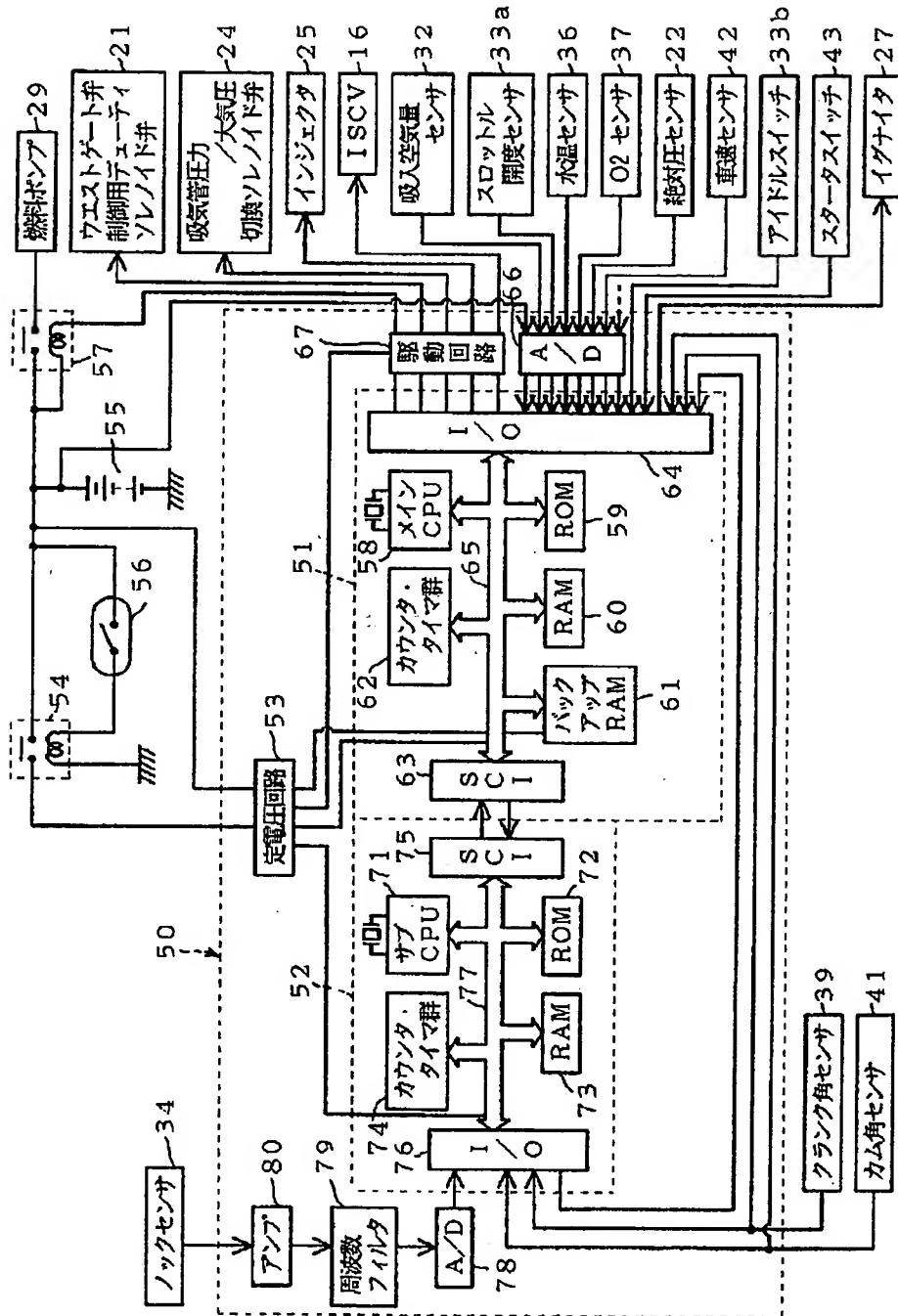
【図20】



【図19】



【図23】



**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☒ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

THIS PAGE BLANK (USPTO)